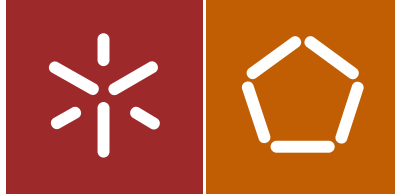


Universidade do Minho  
Escola de Engenharia

Mário Jorge Garcia de Oliveira

Desenvolvimento de uma plataforma  
para Internet das Coisas baseada em Wi-Fi



Universidade do Minho  
Escola de Engenharia

Mário Jorge Garcia de Oliveira

Desenvolvimento de uma plataforma  
para Internet das Coisas baseada em Wi-Fi

Dissertação de Mestrado  
Ciclo de Estudos Integrados Conducentes ao Grau de  
Mestre em Engenharia de Telecomunicações e Informática

Trabalho efetuado sob a orientação do  
Professor Doutor José Augusto Afonso  
Professor Doutor João Luiz Afonso

# Agradecimentos

Queria agradecer a todos os professores e colegas que fizeram parte do meu percurso académico na Universidade do Minho, muito obrigado por todo o conhecimento transmitido ao longo destes anos.

Um especial agradecimento ao Professor Doutor José Augusto Afonso por toda a orientação, paciência e total disponibilidade ao longo de todo o desenvolvimento desta dissertação.

Ao Professor Doutor João Luiz Afonso e ao Pedro Silva muito obrigado por toda a disponibilidade apresentada no desenvolvimento desta dissertação.

Uma menção especial aos meus amigos Carlos Vilela, Gustavo Portela, Mário Costa, André Pereira, Francisco Lima e Fábio Lima, com quem partilhei os melhores momentos ao longo de todo o meu percurso académico um muito obrigado por todo o apoio e amizade transmitida ao longo destes anos.

Um agradecimento a Diana Marques por todo o apoio e compreensão demonstrada ao longo do meu percurso académico.

Por ultimo um agradecimento à minha família, por tornar possível o meu percurso académico e por todo o apoio dado ao longo do mesmo, em especial no desenvolvimento desta dissertação.



## Resumo

O principal objetivo desta dissertação visa o desenvolvimento de uma plataforma genérica para a Internet das Coisas. Para o desenvolvimento da plataforma genérica recorreu-se à utilização de um dispositivo Arduino e de um módulo Wi-Fi. O Arduino tem como objetivo permitir interligar sensores utilizando para isso as interfaces de comunicação disponibilizadas, sendo elas a interface UART, I2C, SPI e ADC.

A interface de comunicação usada para realizar a comunicação entre o Arduino e o módulo Wi-Fi escolhido (ESP8266) é a interface UART. Para a implementação do sistema, desenvolveu-se um *firmware* para o módulo ESP8266, que permite realizar a conexão do módulo Wi-Fi a qualquer rede Wi-Fi presente nas imediações do módulo ESP8266.

Esta dissertação tem um segundo objetivo que foi realizado em parceria com um aluno do Mestrado Integrado em Eletrónica Industrial e Computadores (MIEEIC) cujo tema de dissertação é o desenvolvimento de um sistema de monitorização de consumos e de qualidade de energia elétrica (QEE). Neste segundo objetivo foi desenvolvido um protocolo de comunicação para recolha de dados dos monitores de QEE e uma aplicação para PC (*Personal Computer*) para realizar a apresentação dos valores recolhidos. O sistema usado para realizar a comunicação entre os monitores QEE e a aplicação desenvolvida para o utilizador é a plataforma genérica para a Internet das Coisas.

O projeto realizado nesta dissertação foi concluído com sucesso, sendo comprovado o correto funcionamento da plataforma genérica para a Internet das Coisas. A plataforma desenvolvida permite realizar a devida interação entre os sensores e a aplicação do utilizador. Toda a programação desenvolvida para a plataforma genérica foi devidamente testada, bem como a aplicação desenvolvida

para o utilizador para interação com o monitor de QEE, apresentando bons resultados.

# Abstract

The main objective of this dissertation is the development of a generic platform for the Internet of Things. For the development of the generic platform, an Arduino device and a Wi-Fi module were used. The Arduino device allows the interconnection of sensors using the available communication interfaces, such as UART, I2C, SPI and ADC.

The communication interface used to perform the communication between the Arduino and the chosen Wi-Fi module (ESP8266) is the UART interface. For the implementation of the system, a firmware was developed for the ESP8266 module, which allows the connection between the Wi-Fi module and the Wi-Fi network present near the ESP8266 module.

This dissertation has a second objective that was done in partnership with a student of the Integrated Master's in Industrial Electronics and Computers (MIEEIC) whose dissertation theme is the development of a system for monitoring of consumption and power quality (PQ). In this second objective it was developed a communication protocol for data collection of the PQ monitors and a PC (Personal Computer) application to perform the presentation of the collected values. The system used to communicate between PQ monitors and the application developed for the user is the generic platform for the Internet of Things.

The project done in this dissertation was successfully completed and the correct functioning of the generic platform developed for the Internet of Things was demonstrated. The developed platform allows the interaction between the sensors implemented in the generic platform and the User. All the developed code for the generic platform was properly tested, as well as the application developed for the user for interaction with the PQ monitor, presenting good results.





# Índice de conteúdos

<b>Agradecimentos.....</b>	<b>i</b>
<b>Resumo .....</b>	<b>iii</b>
<b>Abstract.....</b>	<b>v</b>
<b>Índice de conteúdos.....</b>	<b>vii</b>
<b>Lista de figuras.....</b>	<b>xi</b>
<b>Lista de tabelas.....</b>	<b>xv</b>
<b>Acrónimos .....</b>	<b>xvii</b>
<b>1. Introdução .....</b>	<b>1</b>
1.1 Enquadramento e motivação .....	1
1.2 Objetivos .....	6
1.3 Estrutura da dissertação .....	8
<b>2. Revisão Bibliográfica .....</b>	<b>11</b>
2.1 Fundamentos Teóricos.....	11
2.1.1 Wi-Fi.....	11
2.1.1.1 Arquitetura .....	12
2.1.1.2 Camada física .....	13
2.1.1.3 Camada MAC .....	13
2.1.2 Bluetooth .....	15
2.1.3 ZigBee .....	16
2.1.4 Sumário.....	17
2.2 Arduino.....	18

---

<b>3.</b>	<b>Estado da Arte .....</b>	<b>19</b>
3.1	WeMo .....	19
3.2	Canary .....	20
3.2.1	Connected by TCP .....	22
3.3	Neurio .....	23
3.4	ConnectSense.....	24
3.5	Honeywell Total Connect.....	25
3.6	Ivee.....	27
3.7	Sem.se Mother.....	29
<b>4.</b>	<b>Conceção do sistema.....</b>	<b>31</b>
4.1	Arquitetura.....	31
4.2	Bloco Dispositivo .....	32
4.2.1	PC utilizado para emulação do monitor de QEE.....	32
4.2.2	Arduino .....	33
4.2.2.1	Comunicação SPI .....	34
4.2.2.2	Comunicação I2C .....	35
4.2.3	Módulo Wi-Fi ESP8266 .....	37
4.3	Bloco Router.....	41
4.4	Bloco PC do Utilizador.....	42
<b>5.</b>	<b>Implementação do Sistema .....</b>	<b>43</b>
5.1	Implementação da plataforma IoT genérica .....	43
5.1.1	Desenvolvimento do programa do Arduino.....	44
5.1.2	Desenvolvimento do firmware do módulo Wi-Fi .....	46
5.1.2.1	Atualização do firmware do módulo ESP8266 .....	49

5.1.3	Aplicação para teste da plataforma genérica.....	50
5.2	Implementação da aplicação para os monitores de QEE .....	51
5.2.1	Aplicação desenvolvida para o PC do Utilizador .....	55
5.2.2	Aplicação para emular o monitor de QEE .....	58
<b>6.</b>	<b>Testes e Resultados.....</b>	<b>61</b>
6.1	Teste à conexão com o módulo ESP8266 .....	61
6.2	Teste à comunicação da plataforma genérica para IoT.....	63
6.3	Teste à aplicação que emula o monitor de QEE .....	66
6.4	Teste à aplicação para o GEPE .....	69
<b>7.</b>	<b>Conclusões.....</b>	<b>73</b>
	<b>Referências.....</b>	<b>75</b>



## Lista de figuras

Figura 1.1 – Arquitetura IoT baseada em gateway.....	2
Figura 1.2 – Arquitetura IoT com conexão direta à Internet.....	3
Figura 1.3 – Planta de uma habitação com a implementação da plataforma genérica para IoT. ....	5
Figura 1.4 – Diagrama de blocos do monitor de QEE [9].....	7
Figura 2.1 – Trama MAC da norma IEEE 802.11. ....	14
Figura 2.2 – Arduino UNO com indicação dos pinos das comunicações ADC, I2C, SPI e ICSP.....	18
Figura 3.1 – WeMo Insight Switch e aplicação WeMo [17].....	20
Figura 3.2 – Dispositivo Canary All-in-one Security e respetiva aplicação [18].....	21
Figura 3.3 – Dispositivo Icontrol Networks Piper e respetiva aplicação [19]. ....	21
Figura 3.4 – Componentes do sistema Connected by TCP [20].....	22
Figura 3.5 – Dispositivo Neurio e respetiva aplicação [21].....	23
Figura 3.6 – Dispositivo ConnectSense e respetiva aplicação [22].....	24
Figura 3.7 – Aplicação para smartphone e consola Honeywell [23]. ....	25
Figura 3.8 – Iris Hub [24].....	26
Figura 3.9 - Dispositivo Almond+ [25].....	26
Figura 3.10 – Dispositivos Staples Connect e SmartThings Hub [29] [30]. ....	27
Figura 3.11 – Dispositivo Isee [26].....	28
Figura 3.12 – Dispositivo Sem.se Mother [27].....	29
Figura 4.1 – Arquitetura da plataforma genérica. ....	32

Figura 4.2 – PC que substitui os monitores de QEE. ....	33
Figura 4.3 – Modelo de comunicação SPI. ....	34
Figura 4.4 – Parâmetros CPOL, CPHA e ordem dos bits [14]. ....	35
Figura 4.5 – Modelo de comunicação I2C. ....	36
Figura 4.6 – Configuração do endereço e tipo de operação no protocolo I2C. ....	37
Figura 4.7 – Pinout do módulo ESP8266. ....	38
Figura 4.8 – Envio de Comandos AT para Módulo Wi-Fi. ....	40
Figura 4.9 – Router modelo Hitron BVW 3653 Board v2. ....	41
Figura 4.10 – PC que contém a aplicação do utilizador. ....	42
Figura 5.1 – Arquitetura e ligações da plataforma genérica para IoT. ....	44
Figura 5.2 – Fluxograma do programa do Arduino. ....	45
Figura 5.3 – Fluxograma do firmware do módulo ESP8266. ....	47
Figura 5.4 – Menu do Módulo Wi-Fi. ....	48
Figura 5.5 – Ligações para update do firmware do módulo ESP8266. ....	50
Figura 5.6 – Menu da aplicação do PC do Utilizador para testes de comunicação com o Arduino. ....	51
Figura 5.7 – Arquitetura do sistema com a ligação ao monitor de QEE. ....	51
Figura 5.8 – Trama de Controlo. ....	52
Figura 5.9 – Trama de dados enviada pelo monitor de QEE para a tensão e corrente. ....	53
Figura 5.10 – Trama de dados enviada pelo monitor de QEE para os valores da tensão ou da corrente para as três fases. ....	54
Figura 5.11 – Trama de dados relativa aos alarmes do monitor de QEE. ....	54
Figura 5.12 – Conexão com o módulo ESP8266. ....	55

Figura 5.13 – Conexão realizada com sucesso ao módulo Wi-Fi e escolha da interface a utilizar.....	56
Figura 5.14 – Interface com o utilizador. ....	57
Figura 5.15 – Osciloscópio com representação da tensão. ....	58
Figura 5.16 – Fluxograma aplicação para emular o monitor de QEE. ....	59
Figura 6.1 – Redes visualizadas pelo PC do Utilizador. ....	61
Figura 6.2 – Menu da página Web do módulo Wi-Fi. ....	62
Figura 6.3 – Conexão com sucesso à rede desejada.....	63
Figura 6.4 – Aplicação para interação com o módulo Wi-Fi. ....	64
Figura 6.5 – Monitor série da IDE do Arduino. ....	65
Figura 6.6 – Teste à interface de comunicação UART do Arduino. ....	65
Figura 6.7 – Valores recebidos do PC de emulação. ....	69
Figura 6.8 – Forma de onda da tensão e da corrente da fase A.....	70
Figura 6.9 – Resultados do desfasamento, da potência, potência reativa e potência aparente.....	71
Figura 6.10 – Representação do alarme de sobretensão ativo. ....	72
Figura 6.11 – Representação do alarme de sobretensão inativo. ....	72





## Lista de tabelas

Tabela 2.1 – Comparação entre os vários padrões de rede do IEEE 802.11. ....	12
Tabela 2.2 – Comparação entre os protocolos de comunicação Wi-Fi, Bluetooth e ZigBee (adptado de [11]). ....	17
Tabela 5.1 – Comandos de configuração para o Arduino.....	46
Tabela 5.2 – Comandos de controlo. ....	52
Tabela 5.3 – Alarmes dos monitores de QEE. ....	55
Tabela 6.1 – Valores recolhidos pelo ADC referentes à Fase A. ....	67



# Acrónimos

ADC	Analog-to-Digital Converter
AP	Access Point
API	Application Programming Interface
BSS	Basic Service Set
BSA	Basic Service Area
CCA	Clear Channel Assessment
CRC	Cyclic Redundancy Code
DA	Destination Address
DFIR	Diffuse Infrared
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DSSS	Direct Sequence Spread Spectrum
ESS	Extended Service Set
FCS	Frame Check Sequence
FHSS	Frequency Hopping Spread Spectrum
GEPE	Grupo de Eletrônica de Potência e Energia
GFSK	Gaussian Frequency Shift Keying
GPIO	General Purpose Input/Output
I2C	Inter-Integrated Circuit
IBSS	Independent BSS

ISM	Industrial Scientific and Medical
ICSP	In-Circuit Serial Programming
IEEE	Institute of Electrical and Electronics Engineers
IFTTT	If This Then That
IP	Internet Protocol
IoT	Internet of Things
MAC	Medium Access Control
OFDM	Orthogonal Frequency Division Multiplexing
PAN	Personal Area Network
PC	Personal Computer
PCLP	Physical Layer Convergence Procedure
PMD	Physical Medium Dependent
QEE	Qualidade de Energia Elétrica
SA	Source Address
SC	Single Carrier
SPI	Serial Peripheral Interface
STA	Station
TA	Transmitter Address
TCP	Transmission Control Protocol
RA	Receiver Address
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
ROM	Read Only Memory
UART	Universal Asynchronous Receiver/Transmitter

UDP	User Datagram Protocol
WPAN	Wireless Personal Area Network
ZC	ZigBee Coordinator
ZED	ZigBee End Device
ZR	ZigBee Router



# 1.Introdução

## 1.1 Enquadramento e motivação

A integração de sensores e atuadores nos mais diversos equipamentos utilizados no quotidiano humano visa a melhoria da qualidade de vida e da eficiência energética, sendo uma realidade bem vincada nos dias de hoje. A técnica de incorporar sensores nos equipamentos utilizados no dia-a-dia, aliada ao recurso a comunicações baseadas no protocolo TCP/IP, caracteriza a visão mais amplamente adotada do conceito de Internet das Coisas (IoT – Internet of Things). Os analistas preveem que em 2016 estarão ligados 6,4 biliões de dispositivos à IoT, sendo estimado que se liguem todos os dias 5,5 milhões de dispositivos novos, e que em 2020 existam 20,8 biliões de dispositivos interligados, o que representa um aumento de 22 vezes em relação a 2009 [1].

A Internet das Coisas está presente em distintas áreas, entre elas:

- Cidades inteligentes;
- Agricultura;
- Indústria [2];
- Automação residencial;
- Meio ambiente;
- Saúde [3] [4].

Muitas das vezes nem reparamos a quantidade de dispositivos inteligentes que existem a nossa volta e que fazem parte da nossa vida, formando a Internet das Coisas. No capítulo 2 são referenciados e descritos vários dispositivos inteligentes para aplicação residencial.

A Internet das Coisas pode ser estruturada com base em variadas arquiteturas [5]. A seguir são descritas duas das principais arquiteturas para a implementação das IoT, incluindo a arquitetura usada no desenvolvimento deste trabalho.

Na figura 1.1 está representada uma destas arquiteturas. Esta tem como elemento central um *gateway* cujo objetivo principal consiste em proporcionar conexão à Internet para dispositivos embebidos da IoT (e.g., sensores e atuadores) que não implementam o protocolo TCP/IP. Outro objetivo comum é permitir que vários dispositivos com tecnologias de comunicação diferentes (e.g., Wi-Fi, ZigBee, Bluetooth, Z-Wave) consigam comunicar entre si utilizando o *gateway* como intermediário. O bloco Internet representa a rede mundial, que tem a função de interligar todos os blocos implementados, permitindo realizar a interação entre o utilizador, com base em aparelhos como *smartphones* ou portáteis, e os dispositivos embebidos, não sendo necessário assim que este esteja presente no local onde os dispositivos estão implementados. O bloco Serviços (e.g., *cloud* ou servidor) tem a função de armazenar os dados recolhidos pelos dispositivos e proporcionar serviços de IoT associados, utilizando para isso a ligação à Internet. Como exemplo desta arquitetura, na secção 2.2.6 é possível visualizar dispositivos que utilizam um *gateway* para recolher os dados dos dispositivos implementados.

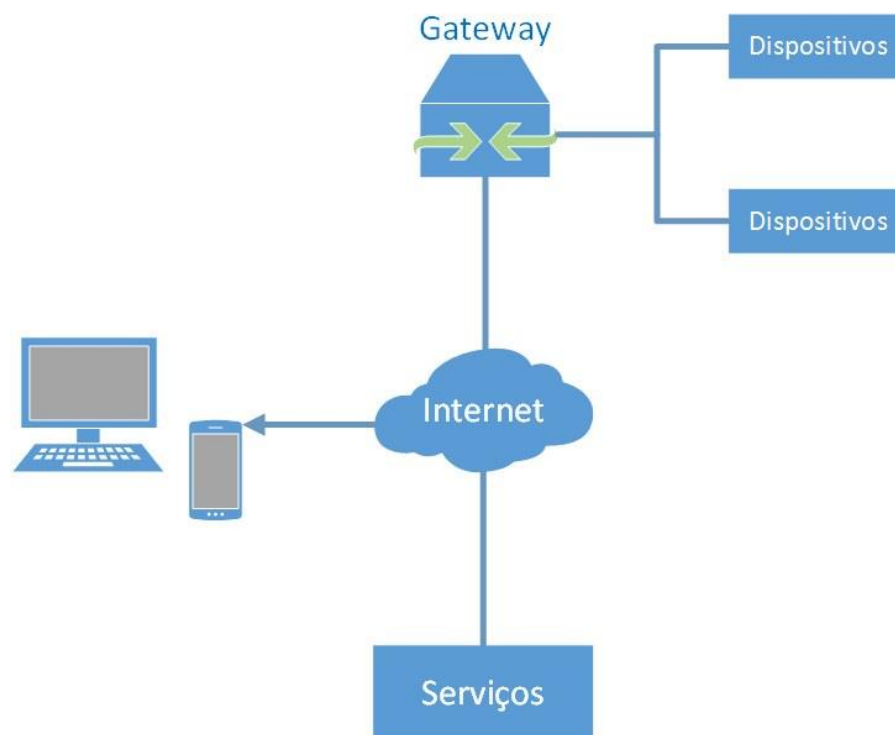


Figura 1.1 – Arquitetura IoT baseada em *gateway*.



A arquitetura apresentada na figura 1.2, em comparação com a arquitetura da figura 1.1, não utiliza um *gateway*, fazendo com que os dispositivos embebidos comuniquem diretamente com a rede de Internet, o que requer que os dispositivos possuam esta capacidade. Neste caso, os dispositivos comunicam diretamente com o *router* de Internet, que por sua vez encaminha os dados recolhidos para o bloco Serviços, permitindo que os dados sejam armazenados numa *cloud* ou servidor e acedidos pelo utilizador utilizando um cliente. Como exemplo, na secção 2.2.1 é possível visualizar um dispositivo da WeMo que não necessita da utilização de um *gateway*.

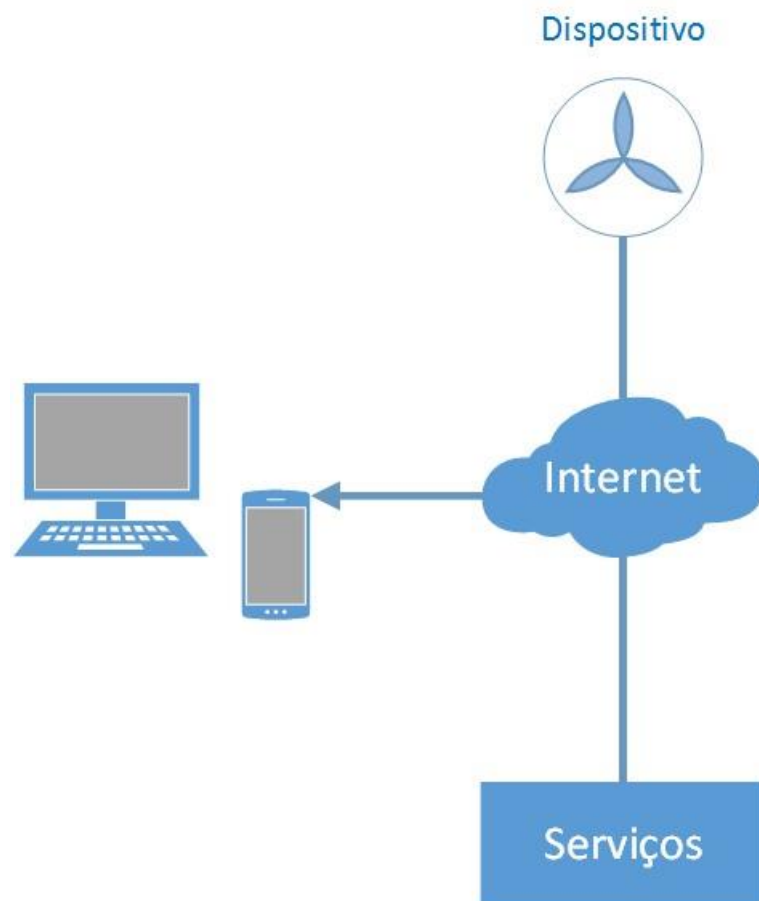


Figura 1.2 – Arquitetura IoT com conexão direta à Internet.

Dentre as vantagens da utilização de um *gateway*, destacam-se as seguintes [5]:

- Possibilita maior segurança no acesso aos dispositivos conectados ao *gateway*, por não haver ligação direta destes à Internet;

- Permite que um vasto número de dispositivos se conecte a ele, fazendo com que não seja necessário conectar cada dispositivo à Internet individualmente;
- Permite a comunicação local com os dispositivos quando não existe comunicação com a Internet, permitindo o correto funcionamento local do sistema;
- Possibilita menor latência na comunicação com os dispositivos quando essa comunicação é feita localmente e não através da Internet;
- Permite que dispositivos com protocolos de comunicação diferentes interajam entre si.

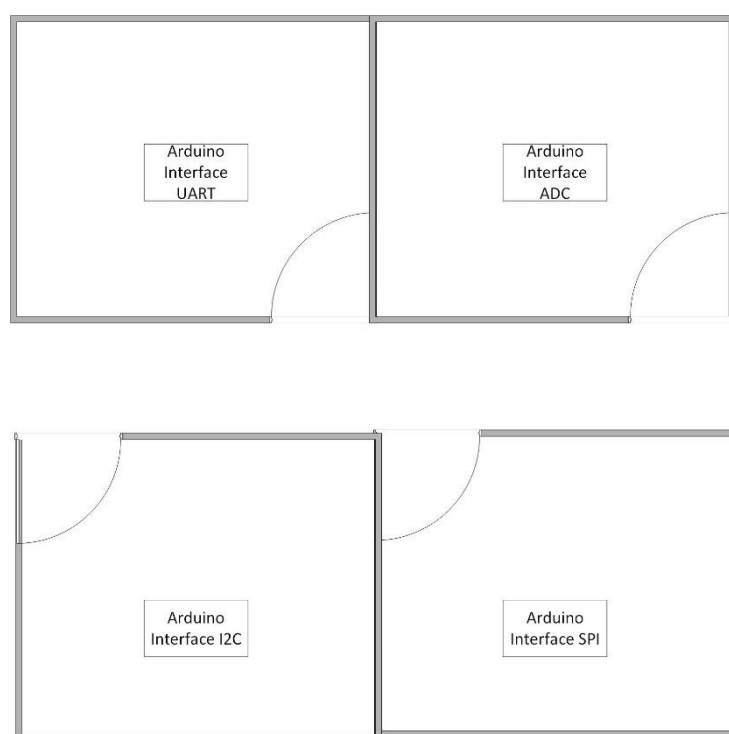
Por outro lado, as desvantagens da utilização de um *gateway* incluem [5]:

- Desenvolvimento mais caro e demorado devido ao facto de não existir um *standard*;
- Criação de mais um ponto de falha do sistema, pois quando o *gateway* deixa de funcionar é perdida a comunicação com os dispositivos;
- Maior latência quando usada a comunicação através da Internet.

Tendo em consideração os prós e contras da utilização de um *gateway*, bem como a aplicação proposta para o sistema (referida a seguir), para o desenvolvimento desta dissertação foi implementada a arquitetura sem *gateway*. Como visto anteriormente, a arquitetura baseada em *gateway* implica a utilização de um equipamento extra, aumenta a complexidade da implementação e tende a aumentar a latência.

Na figura 1.3 está representada um exemplo de uma planta de uma casa, onde ao longo das várias divisões da habitação é implementada a plataforma genérica para a Internet das Coisas desenvolvida nesta dissertação, composta pelo Arduino e o módulo Wi-Fi. O desenvolvimento desta plataforma visa facilitar a implementação de redes IoT e possibilitar a integração de sensores compatíveis com diferentes interfaces (ADC, SPI, I2C ou UART) através de alterações mínimas no código desenvolvido. Sendo assim, a interface de comunicação usada pelo Arduino em cada

divisão da habitação dependerá do sensor/actuador que se deseja usar no local. Por exemplo, com a utilização da interface ADC seria possível controlar as luzes, as tomadas elétricas, etc., com a utilização da interface I2C seria possível usar sensores que permitam medir a temperatura, a humidade, etc., com a interface SPI seria possível usar sensor de impressão digital, etc., a interface UART permitiria usar um determinado sensor de proximidade. A implementação dos respetivos sensores na plataforma genérica para IoT permite automatizar cada divisão da habitação consoante as necessidades de cada pessoa.



**Figura 1.3 – Planta de uma habitação com a implementação da plataforma genérica para IoT.**

Segundo um estudo desenvolvido pela União Europeia (UE), a eficiência energética nas residências pode melhorar em cerca de 13% [6]. A qualidade de energia é degradada por cargas não lineares (e.g., motores elétricos, computadores pessoais, televisores e carregadores), ligadas à rede elétrica, ou seja, existem equipamentos que introduzem deformação na forma de onda (harmónicos). A presença destas cargas, e a consequente poluição harmónica nas instalações elétricas, pode interferir com o bom desempenho da rede e de outros componentes, contribuindo para o aumento de perdas relacionadas com o transporte e distribuição

de energia elétrica, bem como perdas de informação, funcionamentos incorretos, problemas de interferências com sistemas de comunicação, etc.

Uma solução viável para minimizar este problema é a criação de monitores com a capacidade de medir e de informar todas as variações nos parâmetros e consumos existentes na rede elétrica. O mercado atual de monitores de energia elétrica concentra-se principalmente em produtos que permitem ao utilizador obter informações instantâneas dos consumos realizados, oferecendo a possibilidade de calcular o custo total da energia consumida ao longo da semana ou do mês [7].

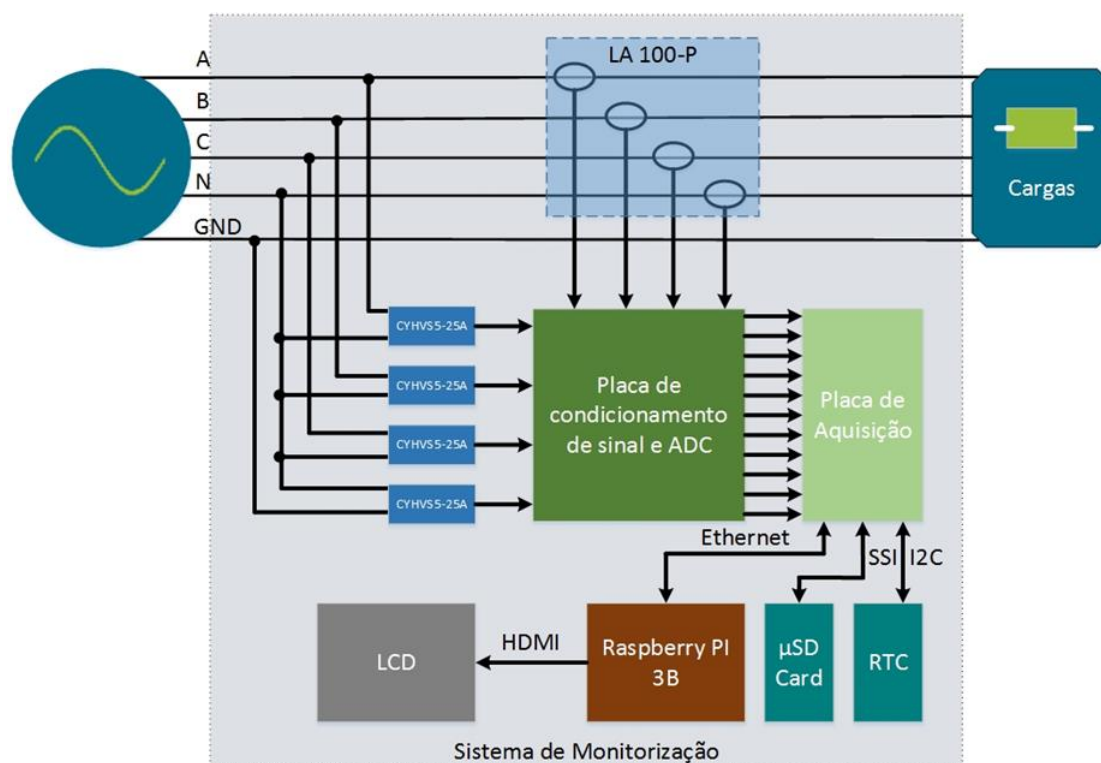
Para a concretização deste projeto, esta dissertação visa também implementar um sistema que seja capaz de recolher, transmitir e apresentar os dados gerados por monitores de energia elétrica num determinado local (e.g., numa habitação, na indústria ou em postos de transformação (PTs) de média tensão para baixa tensão), bem como executar ações sobre os monitores. A aquisição dos valores dos níveis energéticos é feita através de sensores, instalados ao longo do local de implementação, os quais são posteriormente processados por um microcontrolador do monitor e enviados a um PC recorrendo à tecnologia de comunicação sem fios Wi-Fi [8]. Neste sentido, é de extrema importância que o sistema seja capaz de responder em tempo real às exigências, por forma a garantir a qualidade de serviço.

## **1.2 Objetivos**

Um dos objetivos principais desta dissertação é o desenvolvimento de um sistema de uso genérico com a capacidade de realizar a leitura de um vasto número de sensores através de vários tipos de interfaces de comunicação, sendo elas a interface UART, SPI, I2C e ADC. Neste sentido, aquisição dos dados será efetuada por um Arduino UNO, realizando-se a seguir o envio para um módulo Wi-Fi, que por sua vez procederá a transmissão dos dados para o equipamento do utilizador.

O segundo objetivo desta dissertação foi realizado em parceria com um aluno do Mestrado Integrado em Eletrónica Industrial e Computadores (MIEEIC) cujo tema de dissertação foi o desenvolvimento de um sistema capaz monitorizar os consumos e

de qualidade de energia elétrica (QEE) num determinado local. Neste sentido, o trabalho foi realizado em paralelo, sendo atribuídas tarefas distintas aos alunos. A tarefa do aluno do MIEEIC, realizada no âmbito do Grupo de Eletrónica de Potência e Energia (GEPE) da Universidade do Minho, consistiu no desenvolvimento dos monitores de QEE. A Figura 1.4 apresenta o diagrama de blocos do monitor de QEE [9].



**Figura 1.4 – Diagrama de blocos do monitor de QEE [9].**

No contexto deste trabalho conjunto, o segundo objetivo principal desta dissertação consiste em elaborar um protocolo de comunicação para recolha de dados e uma aplicação para PC. Esta aplicação tem como principal função efetuar a apresentação dos dados recolhidos pelos monitores QEE. Para realizar a comunicação entre a aplicação do utilizador e o monitor QEE utiliza-se a plataforma genérica para as IoT desenvolvida neste trabalho. Os monitores também poderão receber informações da aplicação; assim sendo, o sistema desenvolvido usufruirá do recurso a comunicação bidirecional.

Sendo assim, os objetivos gerais para a realização deste projeto são:

- a. Estudo e descrição do Arduino UNO e do módulo Wi-Fi responsável pela comunicação entre a aplicação no PC e os monitores de QEE;
- b. Desenvolvimento do código para a plataforma genérica integrando o Arduino UNO e o módulo Wi-Fi;
- c. Realização de testes de desempenho para garantir o correto funcionamento do sistema de comunicação;
- d. Conceção do protocolo de comunicação usado para a interação entre a aplicação no PC e os monitores;
- e. Desenvolvimento de uma aplicação para PC com a capacidade de realizar a interação com o utilizador e a apresentação dos dados recolhidos pelos monitores de QEE;
- f. Integração e realização de testes com os monitores QEE para garantir o correto funcionamento de todo o sistema desenvolvido.

### **1.3 Estrutura da dissertação**

Esta dissertação está dividida em 6 capítulos:

- No capítulo 1 é feita uma breve introdução ao tema desta dissertação. É abordada a motivação para a realização desta dissertação, bem como os objetivos proposto a alcançar. Por último é realizada a descrição da estrutura da dissertação.
- No capítulo 2 é apresentado o estado da arte das tecnologias de comunicação sem fios, com ênfase nas várias normas da tecnologia Wi-Fi. Por fim são mencionados vários dispositivos que são utilizados na constituição da Internet das Coisas.
- No capítulo 3 são descritos em detalhe a arquitetura e todos os componentes utilizados para o desenvolvimento da plataforma genérica para a Internet das Coisas, com ênfase no Arduino e no módulo Wi-Fi.
- No capítulo 4 é descrito todo o trabalho realizado para a criação da plataforma genérica, descrevendo toda a programação realizada para o

Arduino e o módulo Wi-Fi. Este capítulo contém ainda a descrição detalhada da aplicação desenvolvida para o sistema de monitorização de consumos e de qualidade de energia elétrica do GEPE.

- No capítulo 5 são descritos os vários testes realizados na plataforma genérica para a Internet das Coisas. Os testes realizados têm como principal objetivo testar o protocolo de comunicação desenvolvido para a comunicação entre o utilizador e plataforma genérica. Também são realizados testes à aplicação desenvolvida para o GEPE.
- Por fim o capítulo 6 contém as conclusões obtidas no desenvolvimento desta dissertação bem como a indicação de trabalhos futuros.





## 2.Revisão Bibliográfica

No presente capítulo são abordados os principais temas relacionados com o desenvolvimento desta dissertação. Inicialmente realiza-se um estudo das principais tecnologias de redes sem fios utilizadas atualmente para transmissão de dados de dispositivos da IoT, com especial foco para a tecnologia de comunicação sem fios Wi-Fi, explicando a sua pilha protocolar através da descrição das suas camadas. Por fim é realizado um estudo de trabalhos relacionados, no âmbito da *Internet of Things*, sobre dispositivos inteligentes que utilizem como modo de comunicação uma tecnologia de rede sem fios.

### 2.1 Fundamentos Teóricos

#### 2.1.1 Wi-Fi

A tecnologia de rede precursora às normas IEEE 802.11/Wi-Fi [10] começou a ser desenvolvida, de forma proprietária, no início da década de 90, operando inicialmente na banda de frequência de 900 MHz, disponível na América do Norte. Mais tarde esta tecnologia começou a operar na banda ISM (*Industrial Scientific and Medical*) de 2.4 GHz, disponível globalmente, e para o final da década apareceram produtos a operar na banda de frequências de 5 GHz.

O IEEE aprovou em 1997 as primeiras normas para o IEEE 802.11, permitindo resolver o problema da aderência por parte do mercado, devido ao facto de não existir um padrão que garantisse a compatibilidade entre os dispositivos de diferentes fabricantes. Na tabela 2.1 estão representados alguns dos padrões do IEEE 802.11.

Tabela 2.1 – Comparação entre os vários padrões de rede do IEEE 802.11.

Padrão	Data de lançamento	Banda de frequências	Largura dos canais	Modulação	Tecnologias de antenas	Distância máxima indoor/outdoor (m)	Débito máximo
802.11b	1999	2.4 GHz	20 MHz	DSSS	----	35/140	11 Mbps
802.11a	1999	5 GHz	20 MHz	OFDM	----	35/120	54 Mbps
802.11g	2003	2.4 GHz	20 MHz	DSSS/OFDM	----	38/140	54 Mbps
802.11n	2009	2.4 GHz 5 GHz	20 MHz 40 MHz	OFDM	MIMO	70/250	600 Mbps
802.11ad	2012	60 GHz	2160 MHz	SC OFDM	Beamforming	1/10	6.76 Gbps
802.11ac	2013	5 GHz	40 MHz 80 MHz 160 MHz	OFDM	MIMO MU-MIMO		6.1 Gbps

#### 2.1.1.1 Arquitetura

De acordo com as normas definidas pelo IEEE 802.11, quando um grupo de duas ou mais estações (STA - *Station*) utiliza o mesmo mecanismo de controlo de acesso ao meio (MAC) é formado um conjunto básico de serviço (BSS – *Basic Service Set*). Uma área coberta por uma BSS é designada por área de serviço básico (BSA – *Basic Service Area*).

Uma rede IEEE 802.11 pode operar no modo *ad hoc* ou no modo baseado em infraestrutura. Uma rede *ad hoc* é formada quando existe uma BSS isolada, em que as estações apenas comunicam entre si, sendo denominada por IBSS (*Independent BSS*). Já uma rede baseada em infraestrutura requer uma estação especial, denominada por AP (*Access Point*), servindo de interface entre a BSS e o sistema de distribuição (DS – *Distribution System*), possibilitando a comunicação entre as estações do BSS e entidades externas. O DS permite interligar múltiplos BSS, originando um conjunto alargado de serviço (ESS – *Extended Service Set*).

### 2.1.1.2 Camada física

A camada física do IEEE 802.11 original [10] é baseada em três distintas tecnologias:

- Comunicação ótica (DFIR – *Diffuse Infrared*);
- Espalhamento espectral por sequência direta (DSSS - *Direct Sequence Spread Spectrum*);
- Espalhamento espectral por saltos em frequência (FHSS - *Frequency-hopping Spread Spectrum*).

As últimas duas técnicas operam na banda ISM de 2.4 GHz e a primeira opera na banda dos infravermelhos. Esta norma especifica um débito de transmissão de 1 e 2 Mbps.

A camada física é dividida em duas subcamadas permitindo que a camada física possa operar de forma independente da camada MAC:

- Subcamada PCLP (*Physical Layer Convergence Procedure*);
- Subcamada PMD (*Physical Medium Dependent*).

A camada física do IEEE 802.11 recebe a trama MAC e adiciona o preâmbulo e o cabeçalho da PCLP no início da trama MAC. O preâmbulo e o cabeçalho da camada física são sempre transmitidos à taxa de transmissão de dados mínima da rede, ao contrário do que acontece à trama MAC, que pode utilizar uma taxa de transmissão mais elevada. Isto permite às estações utilizar taxas de transferência de dados diferentes e simplifica a tarefa de equalização. Em contrapartida, isto faz com que o *overhead* da camada física apresente um grande impacto na eficiência de transmissão, sendo mais significativo quanto maior for a taxa de transmissão de dados em comparação com a taxa mínima.

### 2.1.1.3 Camada MAC

O formato da trama de dados que é passada da camada MAC para a camada física é representada na figura 2.2. De referenciar que este é um formato de trama

geral, no caso real alguns campos podem não ser usados. No entanto todos os campos estão presentes nos pacotes de dados.



**Figura 2.1 – Trama MAC da norma IEEE 802.11.**

- O campo de controlo da trama (*frame control*) contém informação que indica o tipo e formato da trama. O IEEE 802.11 define três tipos de trama: dados, controlo e gestão.
- O campo duração/ID contempla dois cenários possíveis: no primeiro cenário indica qual o tempo de duração até a próxima trama.
- Os campos de endereço variam consoante o percurso da trama no sistema de distribuição (DS), podendo ser constituída pelos seguintes tipos de endereço com os seguintes objetivos:
  - Referência à estação de destino da trama (DA- *Destination Address*);
  - Referência à estação de origem da trama (SA – *Source Address*);
  - Referência à próxima estação a receber a trama (RA – *Receiver Address*);
  - Referência ao emissor que está a transmitir a trama (TA – *Transmitter Address*);
  - Referência à BSS em que trama é transmitida (BSSID).
- Campo de controlo de sequência (*sequence control*) é composto por dois subcampos: o número de sequência e o número de fragmento.
- Campo FCS (*Frame Check Sequence*) transporta um código CRC (*Cyclic Redundancy Check*) que deteta os erros na trama, sendo calculado sobre os campos do cabeçalho e do *payload* da trama.

### 2.1.2 Bluetooth

O Bluetooth é uma tecnologia de comunicação de radiofrequência projetada para curto alcance, baixo consumo e dispositivos baratos, com o principal objetivo de substituir os cabos dos dispositivos eletrônicos.

As redes Bluetooth, que são um tipo de rede de área pessoal sem fios (WPAN - *Wireless Personal Area Network*), operam na banda de frequências dos 2.4 GHz, utilizando a técnica de espalhamento espectral por saltos em frequência (FHSS). A banda de frequências utilizada é dividida em 79 canais, separados por 1 MHz entre as frequências das portadoras. A tecnologia Bluetooth utiliza a modulação GFSK (*Gaussian Frequency Shift Keying*) na sua versão original, facultando uma taxa de transmissão de 1 Mbps e com um alcance normal de 10 cm até 10 m, permitindo obter um alcance de até 100 m com o uso de um amplificador de potência.

O Bluetooth apresenta duas topologias de conexão, denominadas por *piconet* e *scatternet*. Uma rede *piconet* é formada por um dispositivo mestre e pelo menos um dispositivo escravo, permitindo a conexão de até sete escravos ao mestre. O mestre da *piconet* é responsável pelo controlo de acesso ao meio, utilizando um protocolo de *polling*. Os dispositivos Bluetooth possuem um endereço único de 48 bits. Através do endereço do mestre é determinada a sequência de saltos, sendo assim atribuída uma sequência única em cada rede *piconet*. Os dispositivos que estão conectados à *piconet* são sincronizados pelo relógio do mestre. A comunicação entre escravos tem que ser feita por intermédio do mestre, ou um desses escravos criar outra rede *piconet*, tornando-se mestre da nova rede criada. A topologia *scatternet* consiste na junção de múltiplas redes *piconet*, sendo que a criação de uma rede *scatternet* é realizada quando um dispositivo faz parte de duas redes *piconet* em simultâneo. Um dispositivo pode ser mestre numa rede *piconet* e escravo numa outra rede, ou escravo em ambas, não podendo ser mestre em duas redes, devido ao facto de que a sequência de saltos seria igual nas duas redes.

### 2.1.3 ZigBee

O ZigBee [11] é uma tecnologia de comunicação do tipo WPAN que disponibiliza baixo consumo de energia e baixas taxas de transferências de dados, permitindo a sua implementação em ambientes de monitorização e controlo no qual os dados são recolhidos por sensores.

O ZigBee possui determinadas características distintas de outras tecnologias sem fios disponíveis no mercado, entre elas, a possibilidade de utilização de topologias de rede em estrela, malha ou árvore.

Os dispositivos ZigBee iniciam a formação de uma PAN (*Personal Area Network*) através do coordenador (ZC – *ZigBee Coordinator*). Os outros dispositivos que se conectam a rede podem ser encaminhadores (ZR – *ZigBee Router*) ou terminais (ZED – *ZigBee End Device*).

O ZigBee utiliza os canais especificados pela norma IEEE 802.15.4, sendo que os canais de 0 a 10 estão definidos para a utilização da banda ISM de 868 MHz, disponível na Europa, e de 900 MHz, disponível nos Estados Unidos, enquanto os canais de 11 a 26 utilizam a banda de 2.4 GHz, disponível globalmente. Na banda dos 2.4 GHz os canais têm um espaçamento de 5 MHz entre eles e disponibilizam uma taxa de transmissão de 250 kbps.

Na formação de uma nova rede ZigBee o coordenador é quem decide o canal que vai ser utilizado, baseado em dois processos:

- Avaliação se o canal está livre através da monitorização passiva do nível de energia;
- Transmissão de um *beacon request* em modo *broadcast* e verificação se algum coordenador responde, para determinar se o canal está livre.

O coordenador ZigBee, com base neste processo, seleciona qual o canal menos utilizado. Para se juntar a uma rede já existente, um nó encaminhador ou terminal transmite um *beacon request*; caso exista uma rede formada no canal, o coordenador transmite um *beacon response*.

### 2.1.4 Sumário

Na tabela 2.2 é apresentada uma síntese sobre as várias tecnologias de comunicação sem fios sendo elas o Wi-Fi, Bluetooth e ZigBee, bem como a descrição das várias normas existentes na comunicação sem fios Wi-Fi.

**Tabela 2.2 – Comparação entre os protocolos de comunicação  
Wi-Fi, Bluetooth e ZigBee (adptado de [11]).**

Standard	Wi-Fi			Bluetooth	ZigBee
Norma IEEE	802.11			802.15.1	802.15.4
	a/b/g/n	ac	ad		
Banda de Frequências	2.4 GHz; 5 GHz	4.6 GHz	60	2.4 GHz	868/915 MHz; 2.4 GHz
Taxa de transferência	11/54/54/600 Mbps	6.1 Gbps	6.67 Gbps	1 Mbps	20/40/250 kbps
Alcance máximo (m)	100 n – 250,	200	10	10	10 -100
Número de canais	14 (2.4 GHz)	37		79	1/10; 16
Largura do canal (MHz)	20 n – 20/40;	40/80/160 MHz	2160 MHz	1	0.3/0.6; 2
Tipo de modulação	BPSK, QPSK, COFDM, CCK, M-QAM	BPSK, 256-QAM	BPSK, 64-QAM	GFSK	BSPK (+ASK), O-QPSK
Espalhamento o espectral	DSSS, CCK, OFDM	OFDM	OFDM	FHSS	DSSS
Célula básica	BSS			Piconet	Estrela
Extensão da célula básica	ESS			Scatternet	Malha, Árvore
Número máximo de nós na célula	2007			8	65000
Encriptação	RC4 stream cipher (WEP), AES block cipher			E0 stream cipher	AES block cipher (CTR, counter mode)
Autenticação	WPA2 (802.11i)			Shared secret	CBC-MAC
Proteção de dados	32-bit CRC			16-bit CRC	16-bit CRC

## 2.2 Arduino

O primeiro Arduino foi desenvolvido na Itália, na cidade de Ivrea, em 2005, pelo professor Massimo Banzi e David Cuartielles. O objetivo de Banzi e Cuatielles era desenvolver um módulo microcontrolador de preço reduzido que permitisse ser usado pelos seus alunos de Arte e Design. A placa foi desenhada por David Cuartielles e programada por David Mellis, aluno de Banziar [13].

Na figura 3.3 é possível visualizar o Arduino UNO, composto por um microcontrolador ATmega328 com memória flash de 32 KB, que permite ser alimentado através da entrada USB ou de uma fonte de corrente contínua (DC). Este módulo oferece uma saída de 3,3 V e outra de 5 V, catorze pinos digitais de entrada/saída e seis entradas analógicas; além disso, seis dos pinos digitais podem ser programados para fornecer uma saída de modulação por largura de pulso (PWM - *Pulse Width Modulation*). Permite o uso de várias interfaces de comunicação, entre elas UART (*Universal Asynchronous Receiver/Transmitter*), SPI (*Serial Peripheral Interface*), I2C (*Inter-Integrated Circuit*) e ADC (*Analog to Digital Converter*). Cada placa inclui um botão de reset e um conector de programação série no circuito (ICSP – *In-Circuit Serial Programming*) [14].

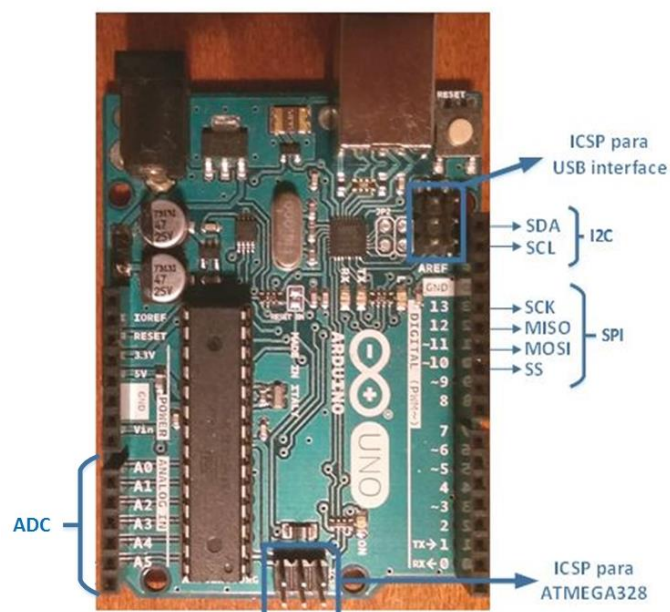


Figura 2.2 – Arduino UNO com indicação dos pinos das comunicações ADC, I2C, SPI e ICSP.



## 3.Estado da Arte

Os objetos inteligentes são os blocos básicos de construção da Internet das Coisas [15]. Em contraste com as etiquetas RFID, os objetos inteligentes contêm algoritmos eficientes e com a capacidade de proporcionar uma interação diferente com os utilizadores, consoante as situações e o meio que os rodeia. Os objetos inteligentes são compostos por várias capacidades distintas, incluindo: sentir; registar; interpretar o que está acontecendo à sua volta; agir por si próprios; comunicar entre si e trocar informação com as pessoas.

Os efeitos mais óbvios da IoT serão visíveis na indústria e nas residências. Neste contexto, a domótica, a vida assistida e a saúde são alguns dos cenários possíveis para este novo paradigma que irá desempenhar um papel de destaque num futuro próximo [16].

Existem atualmente várias empresas com um vasto conjunto de dispositivos inteligentes para a implementação na habitação. Alguns destes dispositivos serão referenciados nas próximas secções, onde é realizada uma breve análise a cada dispositivo.

### 3.1 WeMo

A WeMo Insight Switch consiste numa tomada que tem a funcionalidade de realizar a conexão de eletrodomésticos e aparelhos eletrónicos à rede Wi-Fi, emparelhando luzes, TVs, máquinas de lavar, máquinas de secar, aquecedores, ventiladores e muitos outros eletrodomésticos, e permitindo assim transformar qualquer eletrodoméstico num dispositivo inteligente. Com recurso a um *smartphone* ou *tablet* é possível realizar interações com os vários dispositivos, tais como monitorizar o consumo de energia, ligar ou desligar um dispositivo e receber notificações dos dispositivos ou aparelho que estão a consumir mais energia [17].

Este dispositivo utiliza a tecnologia IFTTT (*If This Then That*), que permite realizar poderosas e fáceis conexões a cada dispositivo WeMo. Esta tecnologia permite criar alertas e conectar os dispositivos a serviços *online*, permitindo criar instruções para ligar as luzes quando o sol se põe, fazer café e abrir as persianas quando acorda, etc. [17]. A figura 3.1 permite visualizar o dispositivo da WeMo e a respetiva aplicação.



Figura 3.1 – WeMo Insight Switch e aplicação WeMo [17].

### 3.2 Canary

O Canary é um sistema de segurança conectado à rede Wi-Fi que realiza a monitorização do espaço onde está inserido com recurso a sensores de movimento, som, temperatura, qualidade do ar e uma câmara de vídeo HD com visão noturna.

A deteção do movimento é feita através de um algoritmo de aprendizagem, ou seja, quanto mais tempo estiver em funcionamento, mais eficaz se torna, permitindo assim enviar notificações mais inteligentes. A figura 3.2 permite visualizar o sistema Canary e a respetiva aplicação. Quando este deteta algum movimento, envia uma notificação, permitindo ao utilizador assistir em direto ou simplesmente escutar o que está a acontecer no local, podendo ativar o alarme e entrar em contacto com as autoridades. Todos os vídeos e áudios são enviados para a *cloud*, que armazena o conteúdo durante 30 dias [18].



**Figura 3.2 – Dispositivo Canary All-in-one Security e respetiva aplicação [18].**

Similar a este sistema existe o Icontrol Networks Piper, que tem a característica de realizar conexão e controlo de dispositivos Z-Wave. Na figura 3.3 é possível visualizar o dispositivo Icontrol Networks Piper [19].



**Figura 3.3 – Dispositivo Icontrol Networks Piper e respetiva aplicação [19].**

### 3.2.1 Connected by TCP

O Connected by TCP é um sistema de iluminação inteligente para a habitação. Inclui um dispositivo *gateway* que utiliza como recurso a tecnologia de comunicação sem fios Wi-Fi para comunicação com as lâmpadas. A conexão com o *router* doméstico é feita através do cabo Ethernet, podendo suportar até 250 lâmpadas conectadas e controlá-las individualmente ou em grupo através de uma aplicação para *smartphone* [20].

O aplicativo móvel permite monitorizar o estado da iluminação, ligar ou desligar as lâmpadas e regular a luminosidade. Este sistema permite pré-programar modos para usar ao longo do dia ou da noite, podendo ligar/desligar a determinadas horas e regular a intensidade da iluminação ao longo do dia. Este sistema atualmente só suporta produtos da sua marca. Na figura 3.4 é possível visualizar o kit do Connected by TCP.



Figura 3.4 – Componentes do sistema Connected by TCP [20].

### 3.3 Neurio

O Neurio monitoriza todo o consumo de energia de uma habitação, usando para isso um algoritmo de aprendizagem para interpretar toda atividade existente, sabendo, por exemplo, quando o utilizador vai para a cama ou quando sai de casa, tendo a capacidade de informá-lo quando algo importante acontece na habitação (por exemplo, quando o forno for ligado é enviada uma mensagem para o utilizador comunicando o evento), e permitindo realizar o controlo dos dispositivos. O Neurio é um dispositivo que se implementa no painel elétrico da habitação e recolhe os consumos de energia de todos os aparelhos ligados a rede elétrica. Com recurso a comunicação Wi-Fi, os dados recolhidos são enviados para um smartphone ou computador. O sistema Neurio está representado na figura 3.5. Este sistema permite visualizar o consumo de energia em tempo real de cada dispositivo, bem como o total de energia consumida durante o dia, a semana, o mês ou o ano, comparar o consumo com os dos vizinhos e obter algumas dicas de como economizar no uso da energia elétrica [21].



Figura 3.5 – Dispositivo Neurio e respetiva aplicação [21].

### 3.4 ConnectSense

O ConnectSense é um sistema de uma empresa que está estabelecida no mercado de sensores para a indústria. Contém sensores de movimento, de luz, de temperatura e humidade, de água e de segurança (e.g., portas, janelas). Cada dispositivo integra um destes sensores referidos. A conexão é feita através da comunicação Wi-Fi, permitindo ao dispositivo conectar-se ao *router* da habitação para enviar o seu estado para a *cloud*, bem como criar alertas para informar o utilizador de quando algo acontece. Alguns sensores usam baterias, sendo ideais para colocar em locais que não tem electricidade, apresentando uma grande mobilidade, podendo ser colocado numa cave, com a finalidade de medir os níveis de humidade. Este sensor tem uma autonomia de tres anos [22]. Na Figura 3.6 é possível visualizar um dispositivo, bem como a respetiva aplicação disponibilizada para o respetivo sensor.

Os sensores comunicam através de Wi-Fi, tornando-se mais fácil a conexão, mas exigindo um maior consumo de energia do que os protocolos usados por outros sistemas para domótica. O ConnectSense pode emitir alertas por chamada de voz, e-mail e texto.



Figura 3.6 – Dispositivo ConnectSense e respetiva aplicação [22].

### 3.5 Honeywell Total Connect

O Honeywell Total Connect Remote Services combina a monitorização de segurança com automação residencial, o que lhe permite monitorizar e controlar uma grande variedade de dispositivos, como câmaras de segurança, alarmes de incêndio, iluminação, janelas, etc. O sistema pode ser monitorizado e controlado a partir do *smartphone*, do computador, ou por meio de uma consola da Honeywell, que permite o controlo por voz (por exemplo, basta dizer “hora de dormir” e automaticamente ativa-se o alarme, apagam-se as luzes, fecham-se as portas) [23]. Na figura 3.7 é possível visualizar o sistema e a sua respetiva aplicação.

Em contraste com o ConnectSense, descrito anteriormente, este sistema não contém qualquer sensor, funcionando apenas como dispositivo central, ou seja, recebe informação dos vários dispositivos sensores colocados ao longo da habitação, disponibilizando esses dados para o utilizador. Este sistema suporta dispositivos Z-Wave, fechaduras da marca Yale e iluminação da Lutron. A Honeywell oferece a sua própria linha de câmaras de segurança. Todos os dispositivos inteligentes são conectados ao dispositivo da Honeywell permitindo assim realizar uma interação com todos os dispositivos inteligentes. Como visto anteriormente para o dispositivo ConnectSense, o Honeywell Total Connect também permite criar alertas quando algo acontece na habitação. A instalação do sistema tem que ser realizada por técnicos



Figura 3.7 – Aplicação para *smartphone* e consola Honeywell [23].

especializados.

A Iris Lowes Hub é um sistema equivalente ao sistema da Honeywell que suporta uma vasta gama de dispositivos e aparelhos inteligentes, desde sensores de segurança até sensores de água. O serviço básico é gratuito e o serviço *premium* custa 9,99 dólares por mês. Este segundo serviço permite criar regras personalizadas, usar o controlo de voz Isee, realizar gravação de vídeo para a nuvem, e enviar alertas para várias pessoas. Tudo é controlado e visualizado pela aplicação para *smartphone* [24]. Na figura 3.8 é possível visualizar o dispositivo Iris Hub.



Figura 3.8 – Iris Hub [24].

O Almond+ é um dispositivo semelhante aos descritos anteriormente, com a diferença de poder funcionar como um *access point* ou aumentar o alcance do *wireless router*. Este dispositivo opera em duas bandas de frequência de Wi-Fi, 2.4 GHz e 5 GHz, e consegue conectar-se a dispositivos que utilizam a tecnologia ZigBee, Z-Wave ou Wi-Fi [25]. O dispositivo pode ser visualizado na figura 3.9.



Figura 3.9 - Dispositivo Almond+ [25].



O Staples Connected Home e o SmartThings Hub oferecem os seus próprios sistemas de domótica, que permitem controlar a casa ou empresa esteja o utilizador lá ou não. Cada kit inclui um *hub* inteligente e uma aplicação para *smartphone* e *tablet*. Estes dois sistemas permitem monitorizar e controlar alarmes de incêndio, termostatos, iluminação, persianas e outros dispositivos, utilizando os protocolos de comunicação ZigBee e Z-Wave. Estes dispositivos permitem conectar dispositivos inteligentes de várias marcas, sendo algumas marcas comuns aos dois dispositivos. Na figura 3.10 é possível visualizar estes dois dispositivos.



Figura 3.10 – Dispositivos Staples Connect e SmartThings Hub [29] [30].

### 3.6 Isee

A Isee é um dispositivo que se conecta aos dispositivos inteligentes implementados na habitação, permitindo realizar a conexão a produtos de várias marcas, tais como, a Nest, WeMo, SmartThings, Philips Hue e da Logitech. A Isee também realiza a conexão à Internet, podendo usufruir de um conjunto de serviços como o Spotify e a Uber. Utiliza a tecnologia de comunicação Wi-Fi, permitindo realizar a comunicação entre os vários dispositivos compatíveis com a Isee implementados na habitação [26].

O dispositivo Iveen funciona através de comandos por voz, ou seja, é apenas necessário realizar um comando direcionado para a Iveen que ela interpreta, usando um processamento de voz avançado, bem como algoritmos próprios para melhorar a precisão, permitindo falar naturalmente para a Iveen como se estivesse a falar para uma pessoa (por exemplo, “Ok Iveen, defina o Nest para 23 °C”). Este dispositivo permite escutar tudo à sua volta até uma distância de cinco metros.

A Iveen não só escuta como também responde usando para isso um microfone omnidirecional. O utilizador pode, por exemplo, pedir à Iveen para “Reproduzir Michael Jackson” usando o serviço Spotify.

A Iveen usa uma aplicação móvel que permite realizar a sua configuração. O aplicativo permite que o utilizador personalize a Iveen e seus dispositivos e serviços favoritos (por exemplo, com o comando "Reproduzir filme Iveen", a TV liga-se, as luzes apagam-se e a temperatura diminui). Na figura 3.11 é possível visualizar o dispositivo Iveen.



Figura 3.11 – Dispositivo Iveen [26].

### 3.7 Sem.se Mother

O Sem.se Mother inclui um *hub* (“mother”) e dispositivos sensores sem fios (“cookies”), que podem ser anexados a vários objetos ou pessoas, permitindo realizar a monitorização dos movimentos, temperatura e localização. Na figura 3.12 é possível visualizar o dispositivo e os respetivos sensores.

Os dispositivos sensores incluem baterias e podem ser colocados em qualquer objeto, por exemplo, nas portas, para monitorizar invasões, ou nas escovas de dentes, para garantir que as crianças andam a escovar os dentes com regularidade. O histórico da atividade é disponibilizado *online* e pode ser configurado para enviar alertas. O *hub* não funciona com dispositivos de outros fabricantes. Os *cookies* contêm baterias que podem durar um ano, caso os *cookies* fiquem perto do *hub*, ou durante meses no caso de estarem distantes do *hub* [27].



Figura 3.12 – Dispositivo Sem.se Mother [27].



## 4. Conceção do sistema

O presente capítulo começa por apresentar a arquitetura do sistema implementado, apresentando de seguida uma descrição dos componentes utilizados, nomeadamente as características do Arduino e do módulo Wi-Fi usados na implementação de todo o sistema elaborado para realização da plataforma genérica e para a comunicação com os monitores QEE desenvolvidos no GEPE.

### 4.1 Arquitetura

A arquitetura implementada para a realização do projeto está representada na figura 4.1, sendo composta por três blocos. O bloco Dispositivo é composto pela conexão entre um Arduino UNO e um módulo Wi-Fi ESP8266, permitindo assim conectar os sensores ao Arduino que realiza a recolha dos dados através das interfaces de comunicação disponibilizadas pelo Arduino UNO. Posteriormente os dados recolhidos são transmitidos para o módulo ESP8266, que realiza a comunicação com o bloco Router utilizando a tecnologia sem fios Wi-Fi e o protocolo de comunicação TCP/IP. O bloco Router proporciona acesso à Internet, que permite enviar armazenar os dados num servidor. Com o acesso à Internet também é possível ter uma interação com os dispositivos estando o PC do Utilizador presente ou não no local, permitindo assim uma maior e melhor interação com os dispositivos implementados com o recurso à plataforma genérica para IoT. Por fim, esses dados são transmitidos para o bloco PC do Utilizador, permitindo ao utilizador visualizar os dados recolhidos no bloco Dispositivo com o recurso a um PC.

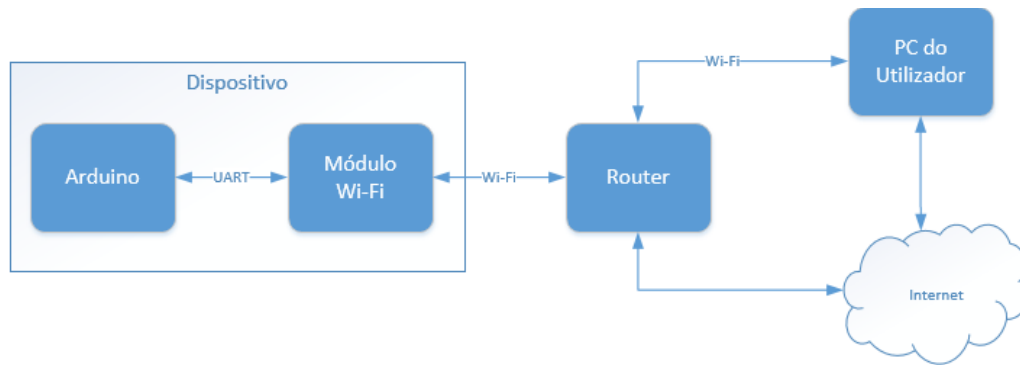


Figura 4.1 – Arquitetura da plataforma genérica.

A arquitetura implementada é similar à arquitetura descrita na figura 1.2, sendo que não recorre à utilização de um bloco de acesso à Internet. Este bloco no desenvolvimento deste projeto não é necessário, sendo que todos os dados observados pelo utilizador são em tempo real, não necessitando de recorrer ao armazenamento de dados na *cloud* ou num servidor. No desenvolvimento da plataforma genérica para IoT foi escolhido a arquitetura sem *gateway*, porque todos os sensores que se desejam implementar são conectados ao Arduino, que por sua vez está conectado ao módulo Wi-Fi utilizando a interface de comunicação UART. Como a única tecnologia sem fios utilizado é o Wi-Fi a utilização de um *gateway* não se torna necessária.

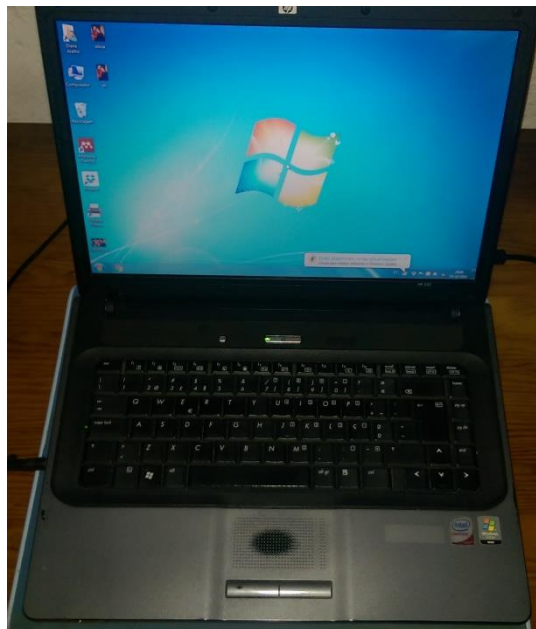
## 4.2 Bloco Dispositivo

### 4.2.1 PC utilizado para emulação do monitor de QEE

Na figura 4.2 está representado um dos PCs utilizado no desenvolvimento desta dissertação. O uso deste PC tem como principal objetivo emular o papel do monitor QEE do GEPE. Este PC está conectado ao Arduino utilizando a porta série, para estabelecer a comunicação entre ambos. Este PC executa a aplicação descrita na secção 5.2.3, que permite realizar o envio para o Arduino de dados idênticos aos que serão enviados pelos monitores de QEE.

O PC utilizado apresenta as seguintes características:

- Windows 7 Profissional 32 bits;
- Processador Intel Dual Core Centrino;
- Memória RAM de 2 GB;
- Disco Rígido de 128 GB;
- Placa Gráfica Medial Accelerator;
- Placa de rede Wi-Fi 802.11 b/g;
- Duas portas USB;
- Uma porta VGA.



**Figura 4.2 – PC que substitui os monitores de QEE.**

#### **4.2.2 Arduino**

O módulo Arduino selecionado para a realização deste projeto de dissertação é o Arduino UNO. Este módulo é responsável por recolher dados dos sensores e transmiti-los para o módulo Wi-Fi, que por sua vez os encaminha para um PC. Por outro lado, também é responsável pelo controlo, o que significa que está habilitado a receber informação transmitida por um PC, executando o comando que lhe foi transmitido.

#### 4.2.2.1 Comunicação SPI

O protocolo SPI (*Serial Peripheral Interface*) é um protocolo de comunicação série síncrona entre microcontroladores e periféricos. O SPI adota um tipo de comunicação mestre/escravo e opera no modo *full-duplex* (um canal para envio e outro para receção). Este protocolo utiliza quatro sinais lógicos, em que dois são de dados e outros dois de controlo. Os primeiros são o MISO (*Master Input Slave Output*) e o MOSI (*Master Output Slave Input*), e os segundos SS (*Slave Select*) e CLK (*Clock*). Através da comunicação *full-duplex*, o mestre pode realizar o envio de dados para um escravo e receber dados dele simultaneamente. Usando o sinal SS, o mestre pode seleccionar com qual dos escravos deseja comunicar [14]. A comunicação SPI é estabelecida através da configuração dos seguintes parâmetros:

- Modo de operação (mestre ou escravo);
- Fase do relógio (normal ou invertida);
- Polaridade do relógio (positiva ou negativa);
- Ordem de transmissão (MSB first ou LSB first);
- Frequência do relógio.

A figura 4.3 ilustra o modelo geral para a comunicação SPI.

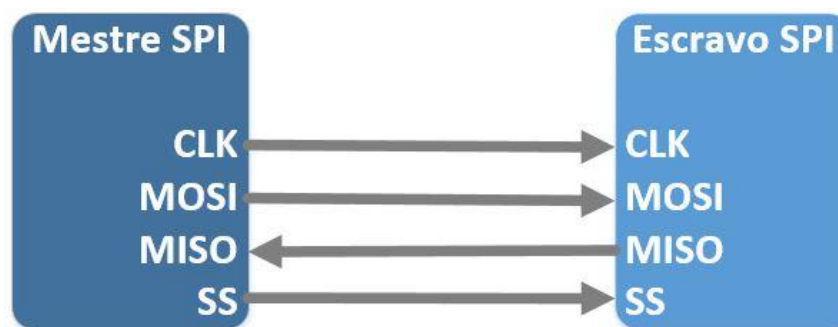


Figura 4.3 – Modelo de comunicação SPI.

Existem quatro configurações possíveis para realizar a configuração da transmissão e a sincronização dos dados entre o transmissor e o recetor usando para isso o sinal do relógio, através da combinação dos parâmetros CPHA (fase do relógio) e CPOL (polaridade do relógio). Quando o CPHA é igual a zero, os dados são lidos na



primeira transição ascendente de clock e os dados são atualizados no barramento na transição descendente; quando igual a um, significa que os dados são capturados na transição descendente do clock e a saída é atualizada na transição ascendente. O CPOL indica a polaridade do sinal quando este está em repouso. Para que haja estabilidade dos dados, o recetor recebe os dados sempre de forma contrária à qual foram transmitidos pelo emissor. A figura 4.4 ilustra as possíveis combinações entre os parâmetros CPOL, CPHA e a ordem de transmissão dos bits [14].

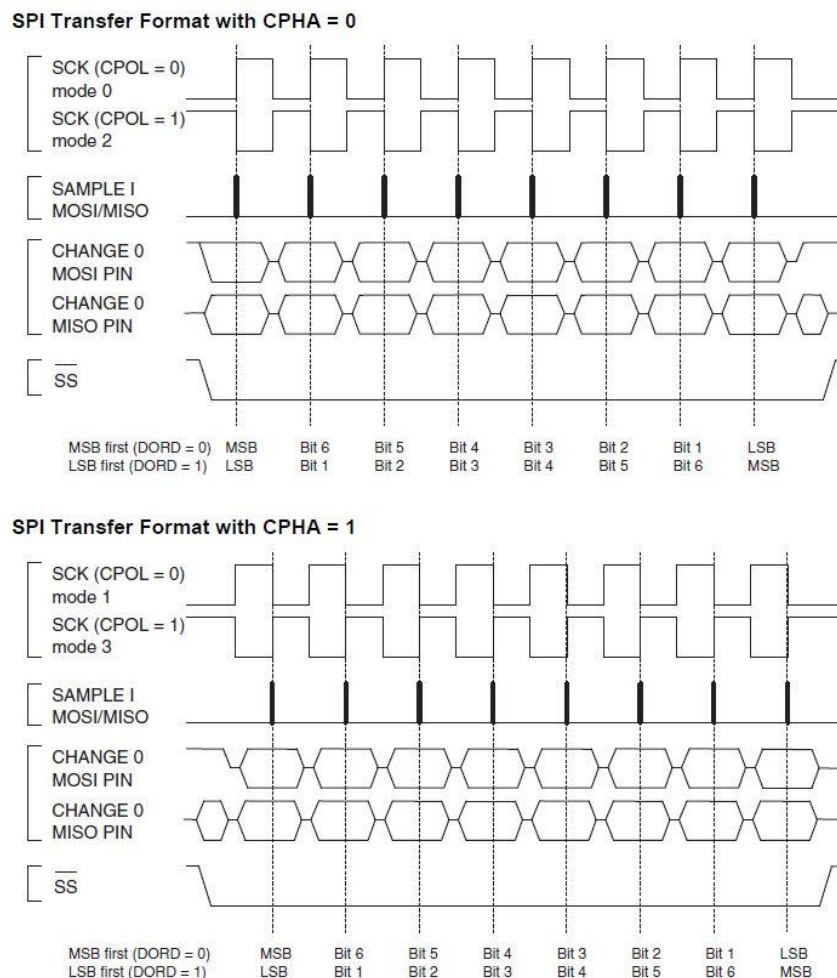


Figura 4.4 – Parâmetros CPOL, CPHA e ordem dos bits [14].

#### 4.2.2.2 Comunicação I2C

Na figura 4.5 esta apresentado o modelo de comunicação do protocolo I2C (*Inter-Integrated Circuit*) é um protocolo de comunicação de barramento; isto significa que, com apenas dois pinos, é possível conectar vários dispositivos,

tornando-se esta a melhor característica deste protocolo. O I2C é semelhante ao protocolo SPI, sendo que também é síncrono e necessita de um mestre e um escravo. A função do mestre é coordenar a comunicação, sendo ele que envia dados ou consulta a informação de determinado escravo. É possível ter mais que um mestre na conexão, e até 128 dispositivos escravos.

Este protocolo utiliza dois pinos lógicos, o SDA (Serial Data) e o SCL (Serial Clock). O SDA é o pino usado para a transferência de dados e o SCL realiza a temporização entre dispositivos, permitindo a comunicação síncrona dos dados na linha SDA. O pino SDA permite realizar o envio e recepção dos dados, sendo assim uma linha bidirecional.

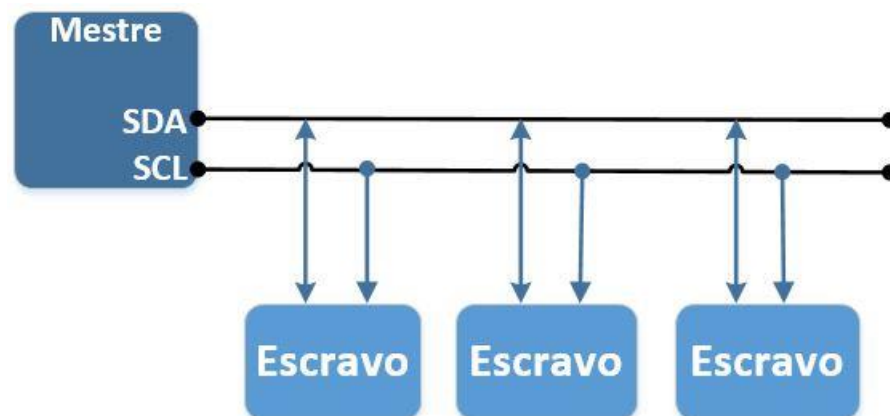


Figura 4.5 – Modelo de comunicação I2C.

Os dispositivos que utilizam o protocolo de comunicação I2C possuem um endereço que os identifica, permitindo assim diferenciar o dispositivo com o qual se deseja comunicar. O endereço utilizado por este protocolo é composto por oito bits, como se pode visualizar na figura 4.6. Os primeiros sete bits indicam o endereço do dispositivo a que se deseja conectar, e o oitavo bit indica qual a operação a realizar, se é de leitura ou escrita. O Arduino, atuando como mestre, configura os sete bits de endereço no início da transmissão do byte e ajusta o oitavo bit consoante a operação desejada.

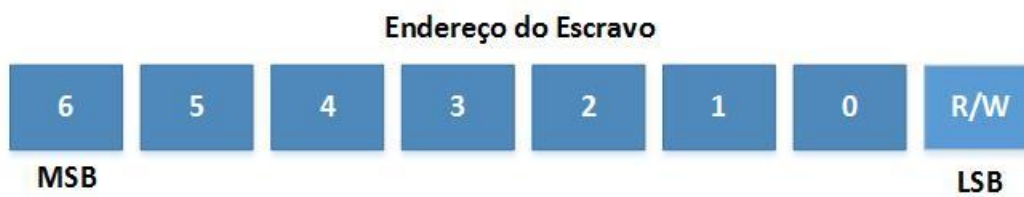


Figura 4.6 – Configuração do endereço e tipo de operação no protocolo I2C.

### 4.2.3 Módulo Wi-Fi ESP8266

O módulo ESP8266 é um dos módulos Wi-Fi disponíveis no mercado. Este módulo se destaca para o desenvolvimento da Internet das Coisas (IoT) devido ao seu baixo preço e tamanho reduzido, sendo composto pelas seguintes características [8]:

- Contém dois pinos digitais GPIO (General Purpose Input/Output).
- Suporta as interfaces de comunicações UART e I2C, e saída PWM.
- Suporta os protocolos 802.11 b/g/n. Opera nos modos estação e cliente. Tem integrado o protocolo TCP/IP com suporte ao IPv4, operando na banda ISM de 2.4 GHz com suporte a WPA e WPA2.
- A arquitetura implementada no módulo é a arquitetura RISC (Reduced Instruction Set Computer) de 32 bits.
- É composto por três memórias:
  - Uma memória RAM (Random Access Memory) utilizada para instruções com uma capacidade de 32 KB;
  - Uma memória RAM para dados com a capacidade de 96 KB;
  - Uma memória ROM de 64 KB de capacidade utilizada para boot.
- Integra um processador com uma frequência de 80 MHz, com a possibilidade de operar nos 160 MHz.
- Inclui uma memória flash SPI Winbond W25Q40BVNIG de 512 KB.

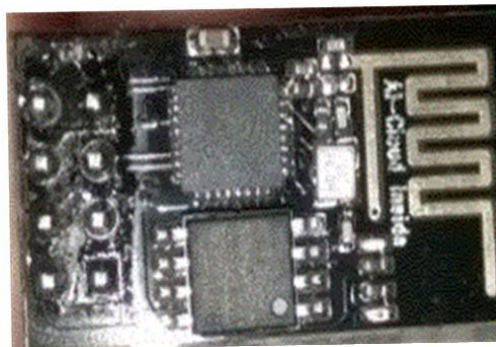
O módulo ESP8266 pode ser utilizado em vários tipos de aplicações, tais como [8]:

- Tomadas inteligentes;

- Automação residencial;
- Monitoramento remoto;
- Controlo via Web de sensores;
- Rede de sensores;
- Controlo industrial.

Na figura 4.7 é possível visualizar o módulo Wi-Fi, bem como a indicação dos seus pinos.

TX	GND
CH_PD	GPIO2
RST	GPIO0
VCC	RX



**Figura 4.7 – Pinout do módulo ESP8266.**

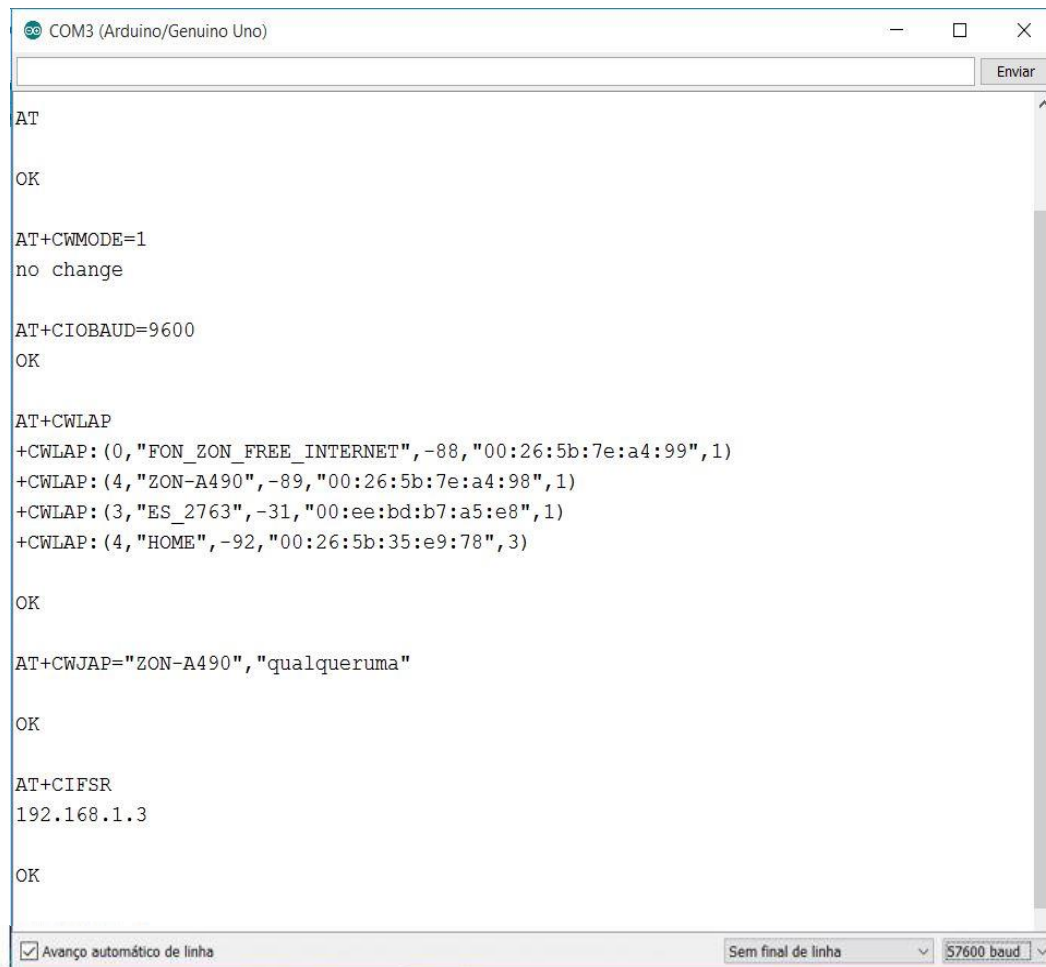
- Vcc – Tensão de alimentação 3,3 V, com um consumo de corrente 300 mA;
- RST – Reset ativado quando ligado ao GND;
- CH\_PD – Sinal de habilitação do módulo, conectado a nível ALTO para operação normal e atualização do firmware;
- TX – Sinal de TX utilizando a comunicação UART, sinal a 3,3 V;
- RX – Sinal de RX utilizando a comunicação UART, sinal a 3,3 V;
- GPIO0 – porta de entrada e saída controlada via firmware, quando ligada ao GND entra no modo update, e para operar no modo normal é ligada a nível ALTO;
- GPIO2 – porta de entrada e saída, permitindo realizar o controlo através do firmware;
- GND – Sinal de massa.

O módulo ESP8266 permite dois tipos de programação, sendo uma delas a programação através dos comandos AT e a outra a programação Lua. A programação

com recurso aos comandos AT consiste no envio de comandos recorrendo à interface UART. Estes comandos variam consoante o *firmware*, sendo que a versão mais atualizada disponível para o módulo é o SDK 1.5.1. No caso da programação Lua, esta consiste na programação de um *firmware* com recurso a bibliotecas disponíveis para o módulo Wi-Fi. O programa desenvolvido é enviado para o microcontrolador do módulo, criando assim um novo *firmware*, e permitindo que o módulo Wi-Fi possa executar tarefas graças ao microprocessador que tem incorporado, não dependendo de outro microcontrolador externo.

Os dois métodos de programação existentes para a interação com o módulo apresentam vantagens e desvantagens, quando comparados. A interação com o módulo ESP8266 através dos comandos AT é o método mais básico para a interação com o módulo. Para realizar o envio dos comandos AT é necessário a utilização de um outro dispositivo por exemplo um módulo Arduino. Quando um comando é enviado para o módulo Wi-Fi, este responde ao pedido realizado enviando a mensagem OK sempre que algo é alterado ou inquirido ao módulo Wi-Fi.

Na figura 4.8 é possível observar o envio de comandos para o módulo com a finalidade de o conectar a uma rede Wi-Fi disponível nas imediações. O envio do comando AT+CWMODE permite selecionar o modo de operação, sendo que o módulo permite operar no modo estação, cliente ou ambos. Quando o comando AT+CWLAP é enviado para o módulo, este devolve todas as redes Wi-Fi disponíveis no alcance. O envio do comando AT+CWLAP="SSID","password", permite selecionar a rede a qual se deseja conectar. Após a conexão do módulo à rede desejada, é enviado o comando AT+CIFSR, que devolve o endereço de IP atribuído ao módulo Wi-Fi na rede a que se conectou.



```

COM3 (Arduino/Genuino Uno)

[Enviar]

AT

OK

AT+CWMODE=1
no change

AT+CIOBAUD=9600
OK

AT+CWLAP
+CWLAP: (0, "FON_ZON_FREE_INTERNET", -88, "00:26:5b:7e:a4:99", 1)
+CWLAP: (4, "ZON-A490", -89, "00:26:5b:7e:a4:98", 1)
+CWLAP: (3, "ES_2763", -31, "00:ee:bd:b7:a5:e8", 1)
+CWLAP: (4, "HOME", -92, "00:26:5b:35:e9:78", 3)

OK

AT+CWLAP="ZON-A490", "qualqueruma"

OK

AT+CIFSR
192.168.1.3

OK

[Avanço automático de linha] [Sem final de linha] [57600 baud]
  
```

**Figura 4.8 – Envio de Comandos AT para Módulo Wi-Fi.**

No caso da programação Lua só é necessário realizar o *upload* do programa desenvolvido para o módulo Wi-Fi, apresentando como limitações a memória disponível no módulo Wi-Fi e o facto que para realizar qualquer alteração no módulo Wi-Fi é necessário alterar o seu *firmware*. O método utilizado para o desenvolvimento deste projeto é com o recurso a programação Lua, permitindo assim que o Arduino não se envolva na operação do módulo Wi-Fi, tendo apenas que realizar o envio dos dados recolhidos dos sensores e a receção dos comandos enviados pelo PC.

### 4.3 Bloco Router

O Hitron BVW 3653 Board v2, representado na figura 4.9 foi o *router* utilizado para intermediar a comunicação entre o PC do utilizador e o módulo Wi-Fi. Este *router*, além de possibilitar o acesso local via Wi-Fi, tem a característica de disponibilizar o acesso à Internet, permitindo testar a comunicação com todo o sistema implementado. Com o acesso à Internet é possível ao utilizador receber os dados recolhidos pelos monitores de QEE e apresenta-los na aplicação desenvolvida para o PC do Utilizador em qualquer local que o utilizador se encontre, permitindo assim ao utilizador ter controlo sobre a plataforma genérica para IoT.



Figura 4.9 – Router modelo Hitron BVW 3653 Board v2.

Este modelo apresenta seguintes características [28]:

- Tensão de entrada 12 V;
- Potencia consumida 12 W;
- Suporta os protocolos de comunicação 802.11 b/g/n;
- Utiliza os protocolos de modulação QPSK, 16QAM, 32QAM, 64QAM e 128QAM;
- Pode operar no modo Access Point;
- Disponibiliza quatro portas RJ45 e duas portas RJ11;
- Inclui uma entrada USB.

## 4.4 Bloco PC do Utilizador

O PC representado na figura 4.10 é o PC responsável por correr a aplicação desenvolvida para o utilizador, permitindo testar a comunicação entre todos os componentes implementados para a formação da plataforma genérica para a Internet das Coisas.

Este PC é composto pelas seguintes características:

- Windows 10 Professional 64 bits;
- Processador Intel i7 2.4 GHz;
- Memória RAM de 4 GB;
- Disco rígido de 256 GB;
- Placa gráfica Intel HD 4000;
- Duas portas USB 3.0;
- Uma porta mini VGA;
- Uma porta micro HDMI.



Figura 4.10 – PC que contém a aplicação do utilizador.



## 5. Implementação do Sistema

Este capítulo está dividido em duas partes principais. A primeira parte descreve a implementação da plataforma genérica para IoT, incluindo o trabalho de programação do Arduino, do módulo Wi-Fi e da aplicação para teste da plataforma, desenvolvida na linguagem de programação Java.

Na segunda parte é descrita a aplicação para PC desenvolvida para apresentação dos dados recolhidos dos monitores de QEE, que faz uso da plataforma genérica para IoT desenvolvida, que permite realizar a comunicação sem fios entre os monitores de QEE e o PC do utilizador.

### 5.1 Implementação da plataforma IoT genérica

Na figura 5.1 estão representadas as ligações realizadas para estabelecer a comunicação entre o módulo Wi-Fi e o Arduino UNO. Os dados recolhidos pelo Arduino são enviados para o módulo Wi-Fi através da interface de comunicação UART. Os pinos 8 e 7 foram definidos no programa do Arduino para realizar a comunicação entre o ESP8266 e o Arduino, sendo que o pino 8 foi definido como o RX no Arduino, que se conecta ao pino TX no módulo Wi-Fi, e o pino 7 do Arduino foi definido como TX, sendo ligado ao RX do módulo.

De notar que não se utilizou os pinos 1 e 2 do Arduino, correspondentes à interface de comunicação UART padrão disponibilizada pelo Arduino, optando por definir outros dois pinos para a realização dessa mesma comunicação, utilizando a biblioteca `SoftwareSerial.h` que permite configurar a interface UART em qualquer dos pinos digitais disponibilizados pelo Arduino. Isto permite deixar os pinos atribuídos à interface de comunicação UART padrão livres para uso futuro, nomeadamente para a comunicação com sensores ou atuadores que venham a ser conectados ao Dispositivo.

O módulo Wi-Fi ESP8266 comunica com o PC do Utilizador utilizando o bloco *Router* como intermediário, como se pode visualizar na figura 5.1.

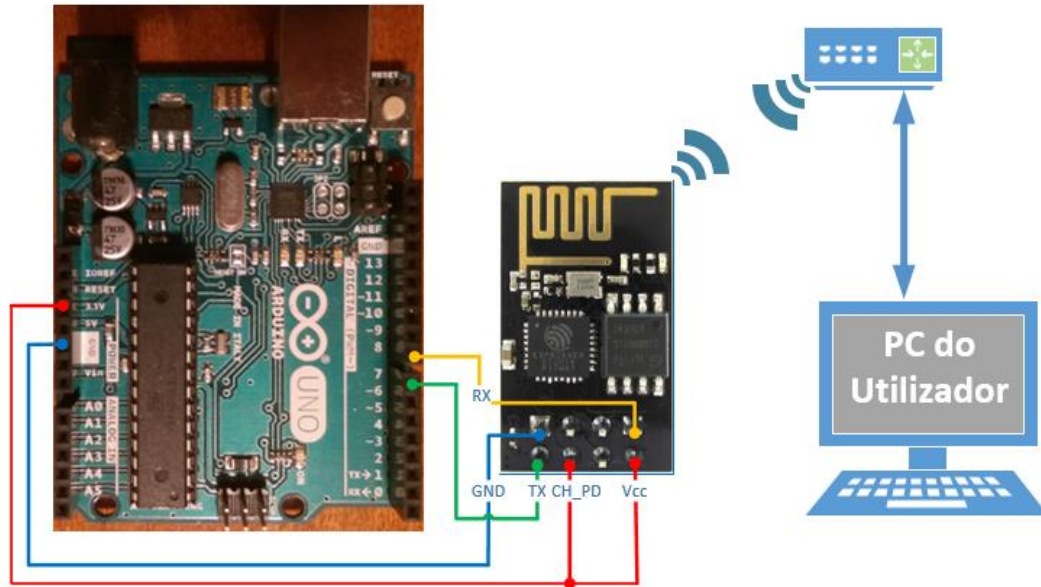
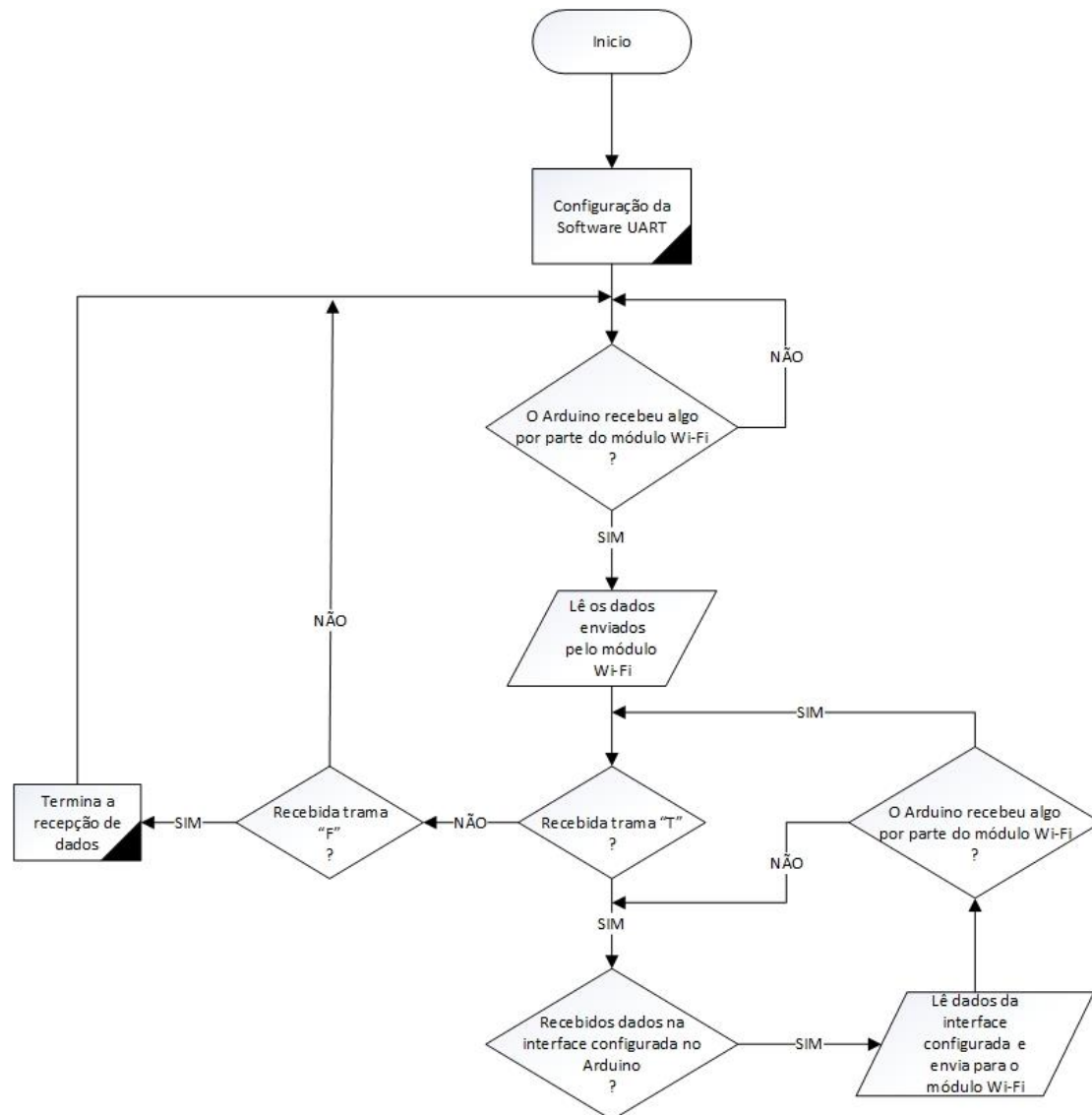


Figura 5.1 – Arquitetura e ligações da plataforma genérica para IoT.

### 5.1.1 Desenvolvimento do programa do Arduino

A programação dos módulos Arduino é desenvolvida com o auxílio de bibliotecas que simplificam a utilização da linguagem C/C++. O IDE do Arduino disponibiliza vários exemplos básicos de código para utilização livre. Nesta dissertação, foi concebida uma solução representada no fluxograma da figura 5.2, que utiliza a biblioteca `SoftwareSerial.h` para comunicação com o módulo Wi-Fi através da interface UART.



**Figura 5.2 – Fluxograma do programa do Arduino.**

O Arduino quando inicia procede à configuração da interface UART por software. São definidos os pinos 7 e 8 do Arduino para uso do RX e do TX e configurada a velocidade de transmissão de 115200 bps.

Realizadas as configurações necessárias para estabelecer a comunicação entre o Arduino e o módulo Wi-Fi, o Arduino fica habilitado a receber dados do módulo Wi-Fi. Enquanto o módulo Wi-Fi não enviar dados para o Arduino este fica sempre à escuta até receber algo por parte do módulo Wi-Fi.

Para adaptar o sistema aos requerimentos específicos dos tipos de sensores utilizados, será necessário realizar pequenas alterações ao código de recolha de

dados de cada sensor através da interface de comunicação disponibilizada pelo mesmo (ADC, SPI, I2C ou UART) quando os dispositivos da plataforma genérica para IoT desenvolvida forem instalados no local (por exemplo, a habitação representada na figura 1.3). Esta alteração no código é facilitada pelas bibliotecas já existentes para configuração e uso das diferentes interfaces de comunicação disponibilizadas pelo Arduino UNO.

Na tabela 5.1 estão definidas as tramas do protocolo de comunicação que permite realizar a ativação do Arduino, as tramas de dados definidas permitem indicar ao Arduino quando deve iniciar ou terminar a leitura de dados por parte da interface de comunicação que for configurada no Arduino, interface esta que será configurada dependendo do sensor que for conectado ao Arduino. A trama T permite definir o intervalo de tempo que o utilizador pretende que o Arduino transmita os dados recolhidos pela interface de comunicação configurada, o Arduino está definido por omissão a transmitir dados 30 em 30 s para o módulo Wi-Fi, se o utilizador definir outro tempo de envio de dados por parte do Arduino a trama T irá conter o tempo que o utilizador pretende receber dados do Arduino (e.g, a trama com o formato T50 indica que o utilizador pretende receber dados 50 em 50 s). A trama F informa o Arduino que deve terminar o envio de dados.

**Tabela 5.1 – Comandos de configuração para o Arduino.**

Trama	Definição
<b>T</b>	Indica ao Arduino para iniciar a leitura de dados, x em x segundos. Por omissão os dados são enviados 30 em 30 s.
<b>F</b>	Indica ao Arduino para desabilitar a interface de comunicação que se encontra ativa no Arduino

### **5.1.2 Desenvolvimento do *firmware* do módulo Wi-Fi**

Como descrito anteriormente, foi necessário desenvolver um *firmware* para o módulo Wi-Fi permitindo assim obter uma melhor performance por parte deste módulo. Recorrendo ao uso dos comandos AT, quando se pretende realizar alguma

alteração no módulo Wi-Fi é necessário enviar os comandos AT para configurar o módulo. Como visto anteriormente, o comando AT tem que ser enviado através da interface UART, não se tornando no melhor método a ser implementado para o desenvolvimento deste projeto, sendo que seria necessário colocar o Arduino a realizar as configurações necessária no módulo Wi-Fi. Para contornar esse problema foi desenvolvido um *firmware* para o módulo Wi-Fi com base no fluxograma da figura 5.3.

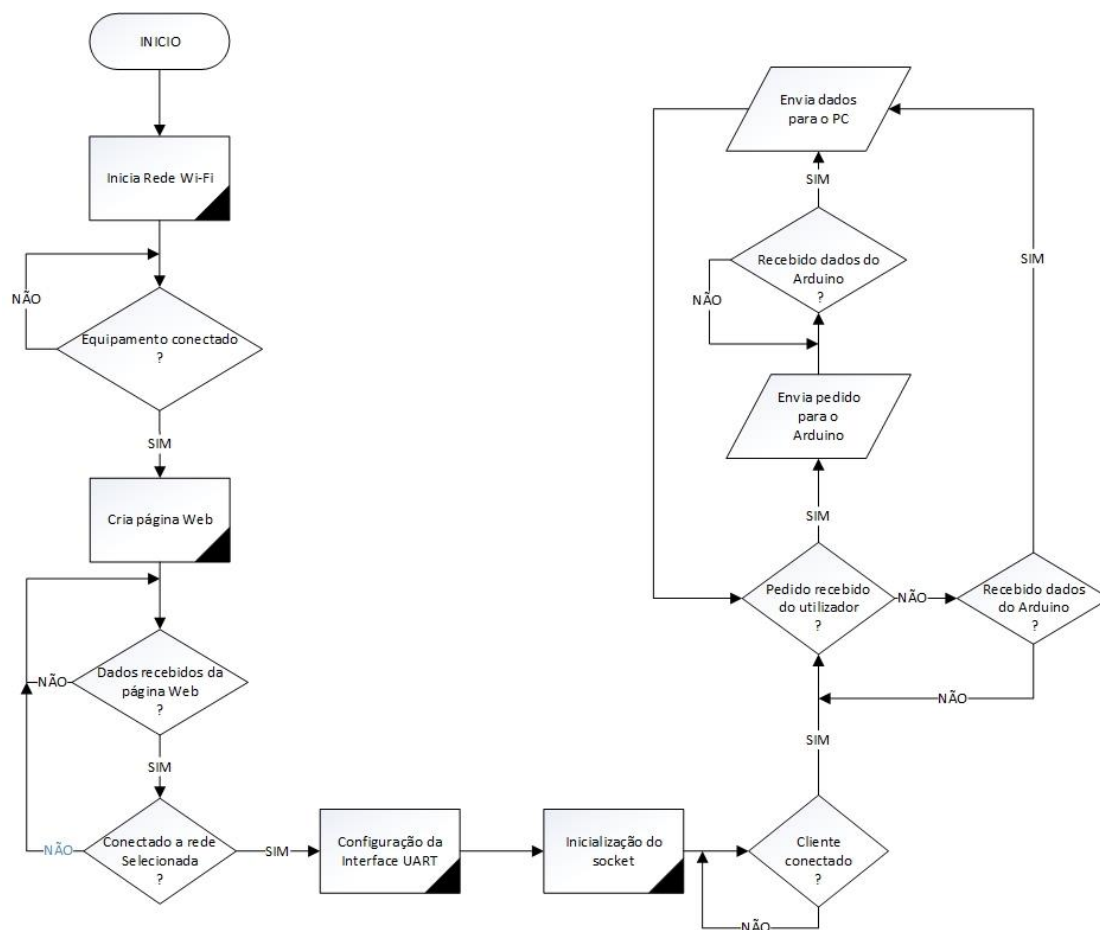


Figura 5.3 – Fluxograma do *firmware* do módulo ESP8266.

No bloco “Inicia rede Wi-Fi” o módulo Wi-Fi é configurado para operar nos modos estação e cliente simultaneamente, permitindo ao módulo Wi-Fi gerar a sua própria rede Wi-Fi e possibilitando a conexão a uma outra rede Wi-Fi nas imediações. O módulo Wi-Fi operando no modo estação oferece a possibilidade a que outros dispositivos que utilizem como recurso a tecnologia Wi-Fi se conectem a

ele. Quando um dispositivo se conecta ao módulo Wi-Fi, é criado um servidor Web no bloco “cria página Web”, com o objetivo de permitir o acesso a uma página Web criada pelo módulo. A página Web disponibilizada pelo módulo Wi-Fi (Figura 5.4) possibilita visualizar as redes Wi-Fi disponíveis nas imediações, permitindo selecionar



Figura 5.4 – Menu do Módulo Wi-Fi.

a rede que se deseja conectar o módulo Wi-Fi, tendo-se que introduzir o SSID e a respetiva palavra-chave da rede desejada.

Posteriormente é configurada a interface de comunicação UART para comunicação com o Arduino, realizada a configuração da interface UART é iniciado um *socket* TCP por parte do módulo Wi-Fi. O *socket* criado tem como objetivo permitir realizar a comunicação entre a aplicação do PC do Utilizador e o módulo Wi-Fi. Iniciado o *socket* o módulo fica à espera que algum cliente se conecte ao servidor (bloco “cliente conectado”). Realizada a conexão entre o servidor e o cliente, o módulo fica habilitado a receber algum pedido por parte do utilizador. Os dados enviados pelo PC do Utilizador são encaminhados para o Arduino através da interface de comunicação UART. A seguir o módulo Wi-Fi fica à espera de receber dados por parte do Arduino. Quando recebidos dados por parte do Arduino, estes são encaminhados para o PC do Utilizador. O módulo Wi-Fi fica constantemente a

verificar se recebeu dados por parte do Arduino ou por parte da aplicação do PC do Utilizador.

As bibliotecas utilizadas para a programação do módulo ESP8266 foram:

- ESP8266WiFi.h esta biblioteca permite realizar a conexão a rede, podendo iniciar servidores, clientes e enviar/receber dados utilizando o protocolo UDP (*User Datagram Protocol*). Também permite utilizar o protocolo DHCP (*Dynamic Host Configuration Protocol*) ou atribuir um endereço IP estático e, por fim, pode utilizar o protocolo DNS (*Domain Name System*);
- A biblioteca WiFiClient.h permite realizar a comunicação com o cliente, permitindo o envio ou receção de dados entre o servidor e o cliente;
- A biblioteca WiFiServer.h permite realizar a inicialização de um servidor utilizando o protocolo de comunicação TCP;
- A biblioteca SoftwareSerial.h possibilita a realização da comunicação UART, permitindo o envio/receção de dados através dessa interface;
- A biblioteca ESP8266WebServer.h permite a criação de um servidor Web, bem a criação de um cliente, permitindo realizar a conexão a um servidor Web e a criação de páginas HTML.

#### **5.1.2.1 Atualização do *firmware* do módulo ESP8266**

Na figura 5.5 está representado o circuito necessário para realizar a atualização do *firmware* para o módulo ESP8266. Para que tudo funcione corretamente é necessário retirar o circuito integrado ATmega328P do Arduino UNO, para que este funcione apenas como FTDI232. Este chip permite realizar a conversão do protocolo USB para o protocolo UART, permitindo assim realizar a comunicação entre o PC e o módulo. Realizando todas as ligações como indicado na figura 4.5 é necessário conectar o pino GPIO0 ao GND e colocar o pino CH\_PD a nível alto, para habilitar o modo de programação do módulo ESP8266. Se tudo estiver a funcionar corretamente o LED azul irá piscar, indicando que o ESP8266 entrou no modo de programação. Feito isto, o módulo Wi-Fi está pronto a receber o novo *firmware*. Para

realizar o desenvolvimento do *firmware* recorreu-se ao programa Arduino IDE versão 1.6.5.

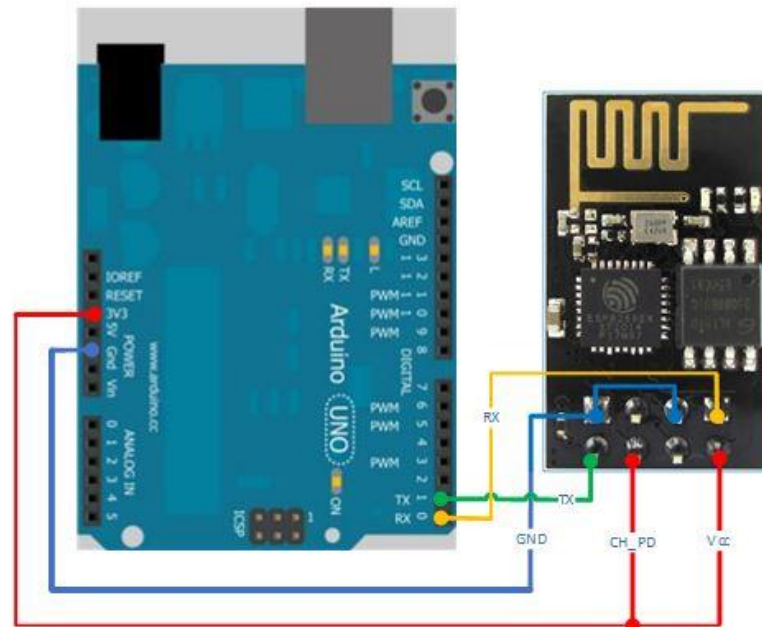


Figura 5.5 – Ligações para *update* do *firmware* do módulo ESP8266.

### 5.1.3 Aplicação para teste da plataforma genérica

Foi desenvolvida uma aplicação com o objetivo de testar a comunicação entre os dispositivos utilizados para o desenvolvimento da plataforma genérica para IoT. A aplicação desenvolvida (Figura 5.6) permite testar o protocolo de comunicação desenvolvido para realizar a ativação da interface de comunicação que se encontra programada no Arduino, tendo também a funcionalidade de permitir receber dados que são enviados pelo Arduino, permitindo assim testar todo o sistema de comunicação entre os vários dispositivos. Esta aplicação permite realizar a conexão ao módulo Wi-Fi através do endereço de IP atribuído ao módulo Wi-Fi e à porta disponibilizada pelo módulo, sendo definida a porta 888. Inseridos estes dados é possível estabelecer a comunicação entre a aplicação e o módulo Wi-Fi. Realizada a comunicação entre o módulo Wi-Fi e aplicação do utilizador é possível enviar dados para o módulo Wi-Fi. Para ativar a interface de comunicação do Arduino é necessário enviar a trama de dados definida para o efeito (Tabela 5.1). Como se pode visualizar



na figura 5.6 existe três possibilidades de interação com o Arduino, com o envio da trama T o Arduino inicia a receção de dados pela interface de comunicação que está programada no Arduino, sendo enviados dados 30 em 30 s, a trama T também permite indicar o intervalo de tempo que se deseja que o Arduino envie dados, para isso é necessário inserir o intervalo de tempo em segundos a seguir ao T (e.g, com T60 o intervalo de tempo é de 60 s), por último a trama F informa o Arduino para desabilitar a utilização da interface de comunicação que se encontrava ativa no Arduino.

```

Output - tese (run) x
run:
Vamos Começar Seja Bem Vindo
Insira o endereço a IP a conectar-se: 192.168.1.2
O cliente se conectou ao servidor!

|-----MENU-----|
| Envie a trama |T| (tempo predefinido de 30 em 30 s para o envio de dados por parte do Arduino). |
| Envie a trama |Tx| (indicando o intervalo de tempo para o envio de dados por parte do Arduino ex: T50). |
| Envie a trama |F| (termina o envio de dados por parte do Arduino). |
|-----|-----|

Escolha a trama de dados a enviar:

```

Figura 5.6 – Menu da aplicação do PC do Utilizador para testes de comunicação com o Arduino.

## 5.2 Implementação da aplicação para os monitores de QEE

Na figura 5.7 está representada a arquitetura implementada para o sistema capaz de monitorizar o consumo e a qualidade de energia elétrica num determinado local, desenvolvida em parceria com um aluno do MIEEIC, que ficou responsável pela recolha de dados através do monitor de QEE. Os dados recolhidos por cada monitor de QEE são enviados posteriormente para o respetivo Arduino utilizando a interface de comunicação SPI.

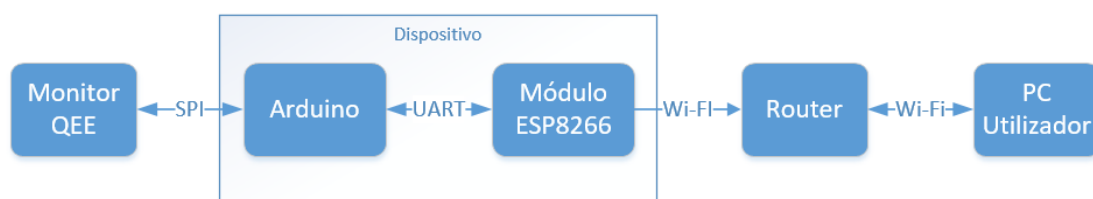


Figura 5.7 – Arquitetura do sistema com a ligação ao monitor de QEE.

Para possibilitar a comunicação entre a aplicação do utilizador e os monitores de QEE do GEPE foi desenvolvido o protocolo de comunicação descrito nesta secção. No sentido da aplicação para o monitor de QEE, foi implementada uma trama de controlo simples, composta por um único campo, que está representada na figura 5.8. Esta trama tem a função de comunicar ao monitor de QEE quais os dados que pretende receber.



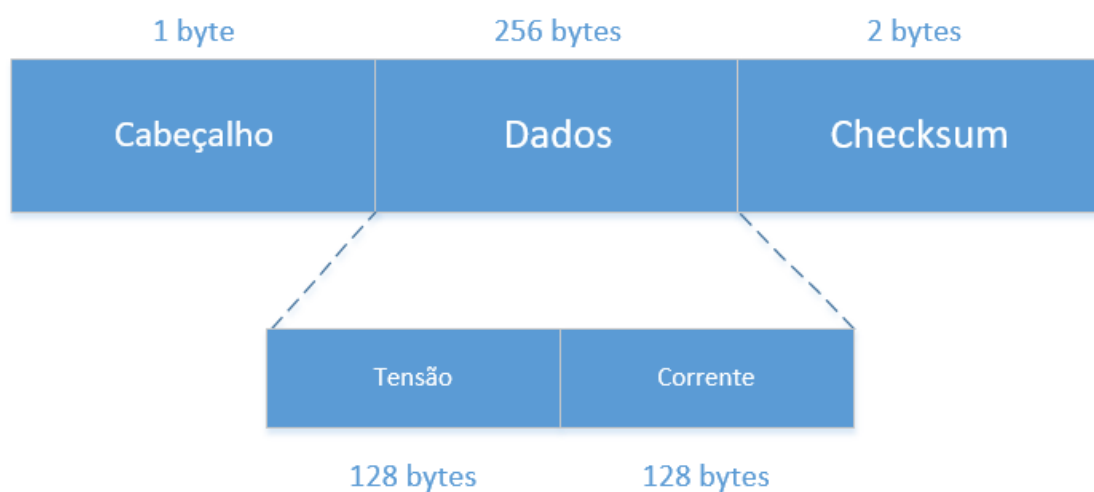
**Figura 5.8 – Trama de Controlo.**

Na tabela 5.2 estão representados os vários comandos disponíveis para realizar a interação com os monitores de QEE

**Tabela 5.2 – Comandos de controlo.**

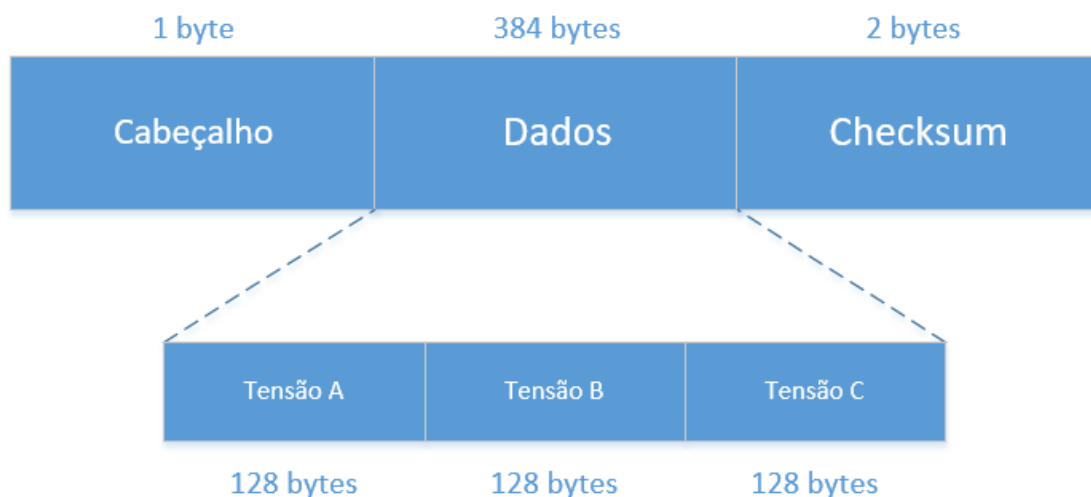
Cabeçalho	
Tipo de TRAMA	DEFINIÇÃO
0	Reservado para uso futuro
1	Ler tensão e corrente da fase A
2	Ler tensão e corrente da fase B
3	Ler tensão e corrente da fase C
4	Ler tensões da fase A, fase B e fase C
5	Ler correntes da fase A, fase B e fase C
7	Reservado para uso futuro
8	Reservado para uso futuro
9	Reservado para uso futuro

Consoante os dados que são pedidos aos monitores de QEE, a trama que é recebida dos monitores de QEE varia. Na figura 5.9 está representada a trama que é enviada pelos monitores de QEE para a aplicação do PC do Utilizador, quando são pedidos os dados da tensão e da corrente, seja da fase A, fase B ou fase C. O cabeçalho indica que tipo de dados que estão a ser transmitidos, tendo o mesmo conteúdo da trama de controlo. No campo Dados é enviada a informação recolhida dos monitores de QEE, que envia 64 amostras do valor de pico da tensão e 64 amostras do valor de pico da corrente relativamente a uma fase, tendo um tempo total de amostragem de 0.019688 segundos com um intervalo de 0.000313 segundos entre cada amostra, sendo que cada amostra tem um tamanho de 2 bytes que representam um inteiro, constituindo no total um *payload* de 256 bytes. O campo Checksum tem a função de conter um valor calculado sobre o campo do cabeçalho e o campo dados, permitindo verificar a integridade dos dados transmitidos. Caso o Checksum não esteja correto os dados são reenviados.



**Figura 5.9 – Trama de dados enviada pelo monitor de QEE para a tensão e corrente.**

Na figura 5.10 está representada a trama de dados enviada pelo monitor de QEE quando são pedidos os dados da tensão ou da corrente nas 3 fases.



**Figura 5.10 – Trama de dados enviada pelo monitor de QEE para os valores da tensão ou da corrente para as três fases.**

Na figura 5.11 está representada a trama que é enviada pelos monitores de QEE quando existe a ocorrência de um alarme. Quando o monitor de QEE detecta um alarme envia a trama com a indicação da ocorrência de um alarme. No cabeçalho é indicado qual o tipo de alarme que ocorreu, podendo ocorrer alarmes de subtensão momentânea, de sobretensão momentânea e de interrupção de energia momentânea. No campo dados é indicado o início ou o fim do alarme enviado pelo monitor de QEE. Na aplicação desenvolvida para o Utilizador ocorre uma transição dos quadrados verdes para vermelhos (secção 5.2.2) indicando assim a ocorrência do alarme referido.



**Figura 5.11 – Trama de dados relativa aos alarmes do monitor de QEE.**

Na tabela 5.3 estão representados os alarmes que podem ocorrer bem como a indicação do tipo de trama, que indica qual alarme está a ocorrer num determinado momento.

Tabela 5.3 – Alarmes dos monitores de QEE.

Cabeçalho	
Tipo de TRAMA	DEFINIÇÃO
10	Alarme de subtensão momentânea
11	Alarme de sobretensão momentânea
12	Alarme de interrupção de energia momentânea
13	Reservado para uso futuro
14	Reservado para uso futuro
15	Reservado para uso futuro
16	Reservado para uso futuro

### 5.2.1 Aplicação desenvolvida para o PC do Utilizador

Na figura 5.12 é apresentada a interface gráfica que permite realizar a conexão com o módulo Wi-Fi introduzindo o endereço IP que lhe foi atribuído quando realizada a conexão à rede.

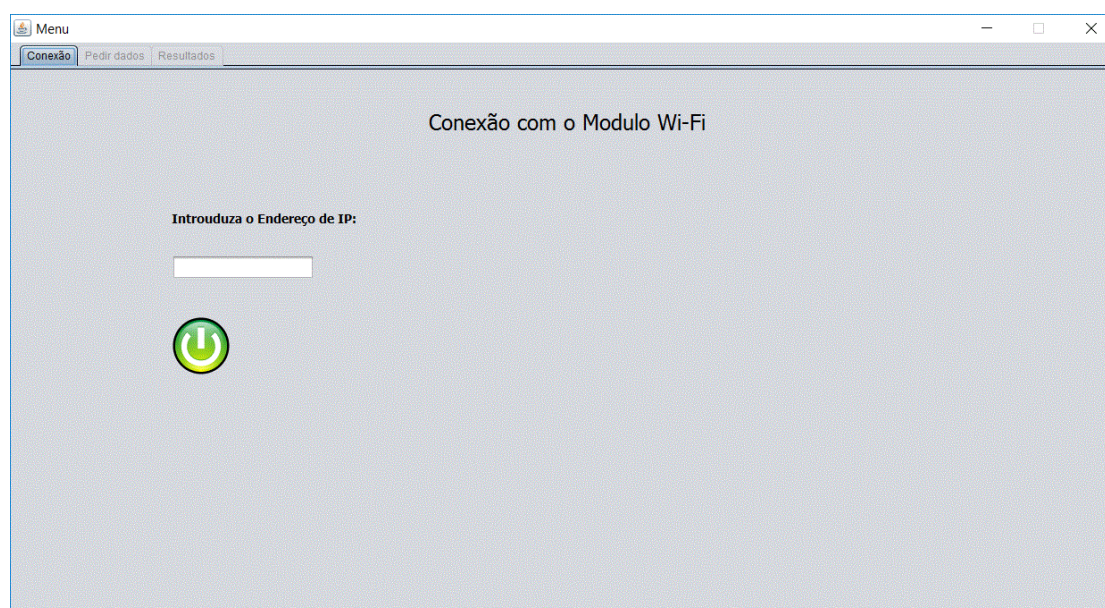
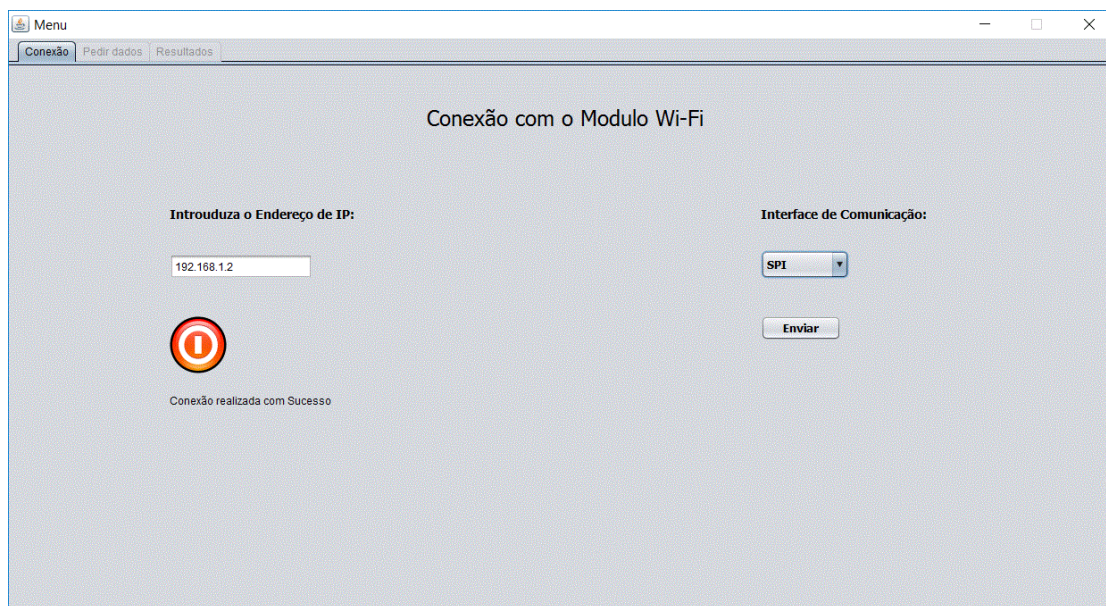


Figura 5.12 – Conexão com o módulo ESP8266.

A figura 5.13 representa a interface gráfica quando a conexão ao módulo Wi-Fi é realizada com sucesso. Nesta janela surge a opção que permite ao utilizador escolher qual a interface de comunicação que pretende ativar no Arduino. A interface de



comunicação selecionada para realizara a comunicação com os monitores QEE é a interface de comunicação SPI.



**Figura 5.13 – Conexão realizada com sucesso ao módulo Wi-Fi e escolha da interface a utilizar.**

Na figura 5.14 estão representadas todas as interações que se pode realizar com o monitor de QEE, permitindo visualizar as tensões e correntes das três fases disponíveis, denominadas por fases A, B e C, bem como obter qualquer tipo de ocorrência que aconteça na linha de transmissão do monitor QEE, podendo ocorrer três tipos diferentes de alarmes:

- O alarme de sobretensão momentânea ocorre quando o valor eficaz da tensão é superior aos 253 V;
- O alarme de subtensão momentânea ocorre quando o valor eficaz é inferior aos 207 V;
- O alarme de interrupção de energia momentânea ocorre quando existe uma falha de energia.

Estes alarmes, quando acontecem, são refletidos na transição dos quadrados verdes para vermelho, indicando a data e hora que o alarme aconteceu e posteriormente quando o alarme deixar de existir é indicada a data e a hora em que o alarme deixou de ocorrer, alertando assim para a presença de um alarme na linha

de transmissão. A aplicação desenvolvida para o PC do Utilizador contém uma *thread* que permite verificar a existência da ocorrência de um alarme detetado pelo monitor de QEE.

Conexão Dados

Escolha a opção que deseja pedir:

☐ Tensão e Corrente da Fase A

☐ Tensão e Corrente da Fase B

☐ Tensão e Corrente da Fase C

☐ Tensão da Fase A, B, C

☐ Corrente da Fase A, B, C

Enviar

Desfasamento: Potência: Potência Reativa: Potência Aparente:

Alarmes sobretensão Data Hora

Alarme de subtensão Data Hora

Alarme de Interrupção Data Hora

Figura 5.14 – Interface com o utilizador.

Quando são recebidos os dados da opção seleccionada (64 amostras) é realizado o tratamento dos dados recebidos, sendo apresentados esses dados num gráfico com o objetivo de simular um osciloscópio (figura 5.15). Recebidas as 64 amostras, também é realizado o cálculo da potência reativa, da potência aparente e da potência total, bem como o cálculo do desfasamento entre as ondas representadas.

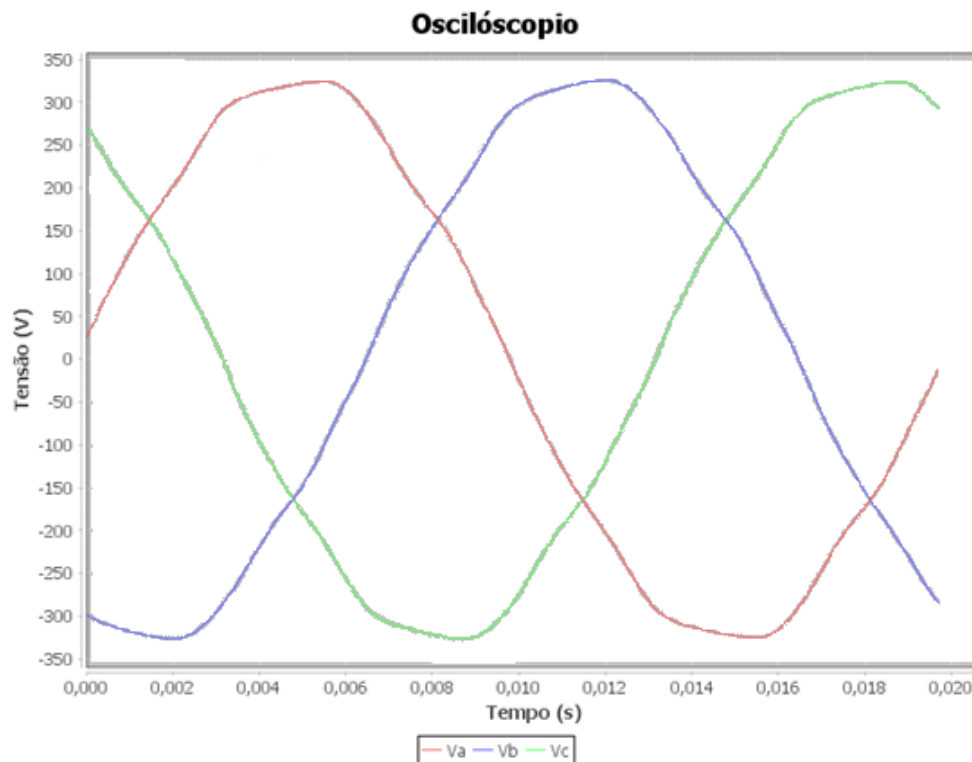


Figura 5.15 – Osciloscópio com representação da tensão.

### 5.2.2 Aplicação para emular o monitor de QEE

A segunda aplicação desenvolvida tem a função de enviar dados idênticos aos dados que serão enviados pelos monitores QEE do GEPE, permitindo assim emular o comportamento de um monitor de QEE com dados reais. A aplicação descrita nesta secção é executada no PC que substitui os monitores de QEE (descrito na secção 4.2.1), utilizando neste caso a interface de comunicação UART para realizar o envio dos dados para o Arduino em lugar da interface SPI. A aplicação de emulação desenvolvida fica à espera de receber a trama de controlo, que indica qual os dados que se pretende receber, caso a trama recebida coincida com um dos valores indicados na tabela 5.2, a aplicação carrega o ficheiro txt correspondente aos dados pedidos pelo Utilizador e realiza o envio desses mesmos dados para o Arduino utilizando a interface de comunicação UART. O Arduino encaminha esses mesmos dados para o módulo Wi-Fi utilizando a outra interface UART, implementada por software, que por sua vez envia os dados para a aplicação para PC desenvolvida para



o GEPE. Na figura 5.16 é apresentando o fluxograma que ilustra a aplicação de emulação desenvolvida.

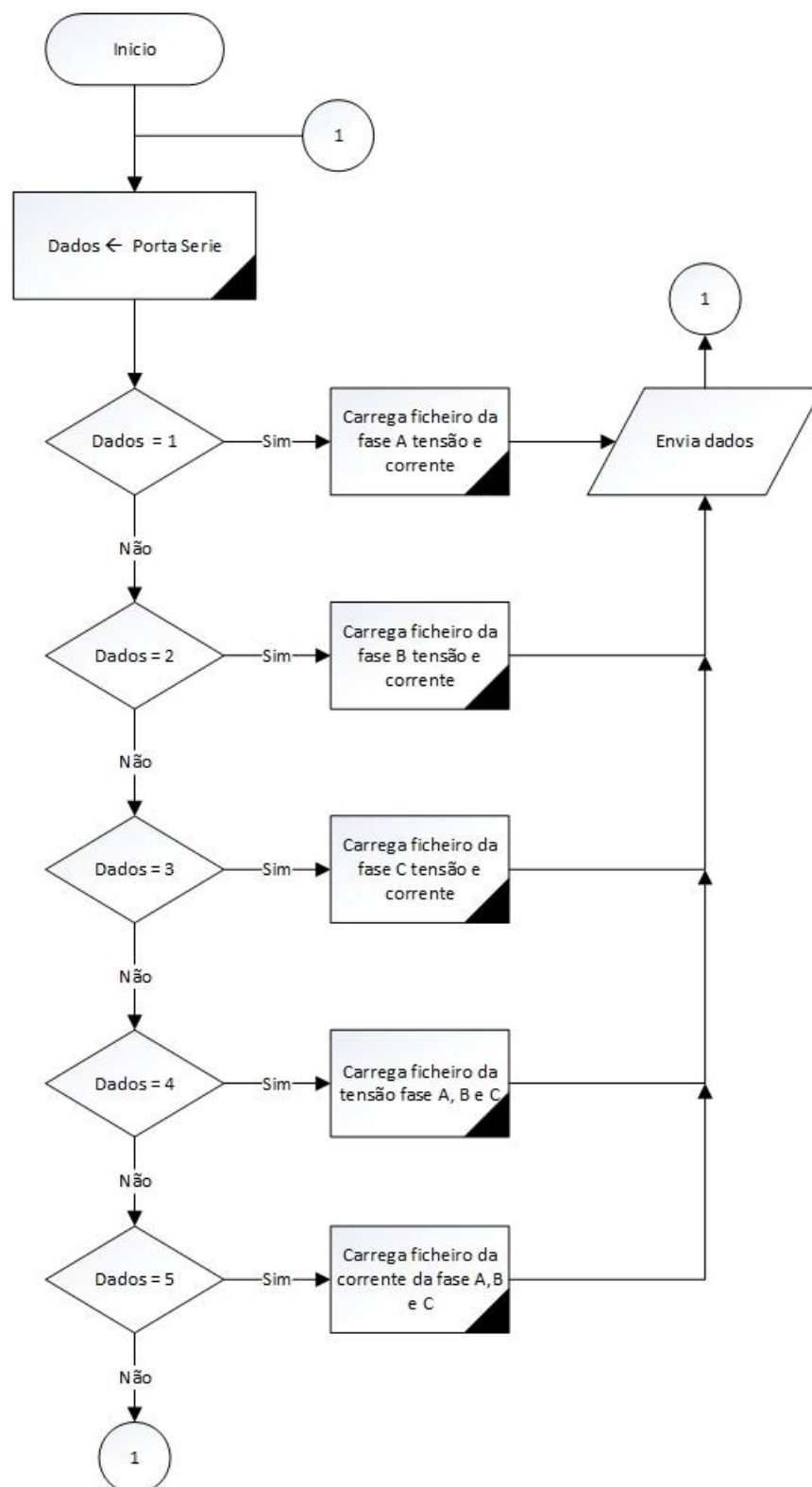


Figura 5.16 – Fluxograma aplicação para emular o monitor de QEE.



## 6. Testes e Resultados

No presente capítulo são apresentados e discutidos os testes realizados para demonstrar o corrente funcionamento do projeto desenvolvido, tendo em conta a comunicação entre os vários componentes implementados no sistema.

### 6.1 Teste à conexão com o módulo ESP8266

Este teste tem como principal objetivo verificar a conexão de um dispositivo que disponha da tecnologia Wi-Fi com a rede gerada pelo módulo Wi-Fi, bem como realizar a conexão do módulo Wi-Fi a uma rede Wi-Fi presente nas imediações.

Neste teste é utilizado o PC do Utilizador (secção 4.4), que permite efetuar a conexão ao módulo Wi-Fi e posteriormente efetuar a conexão do módulo Wi-Fi à rede Wi-Fi gerada pelo Router (secção 4.3). Inicialmente o PC do Utilizador conecta-se ao módulo Wi-Fi através da rede gerada pelo próprio módulo, utilizando o SSID myESP (Figura 6.1). Com a conexão do PC ao módulo Wi-Fi efetuada é possível realizar posteriormente a conexão do módulo Wi-Fi à rede privada do utilizador. Neste caso a rede utilizada tem atribuído o SSID ZON-A490.

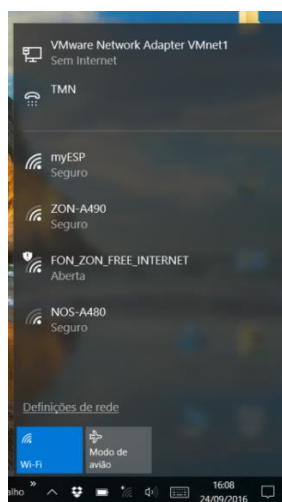
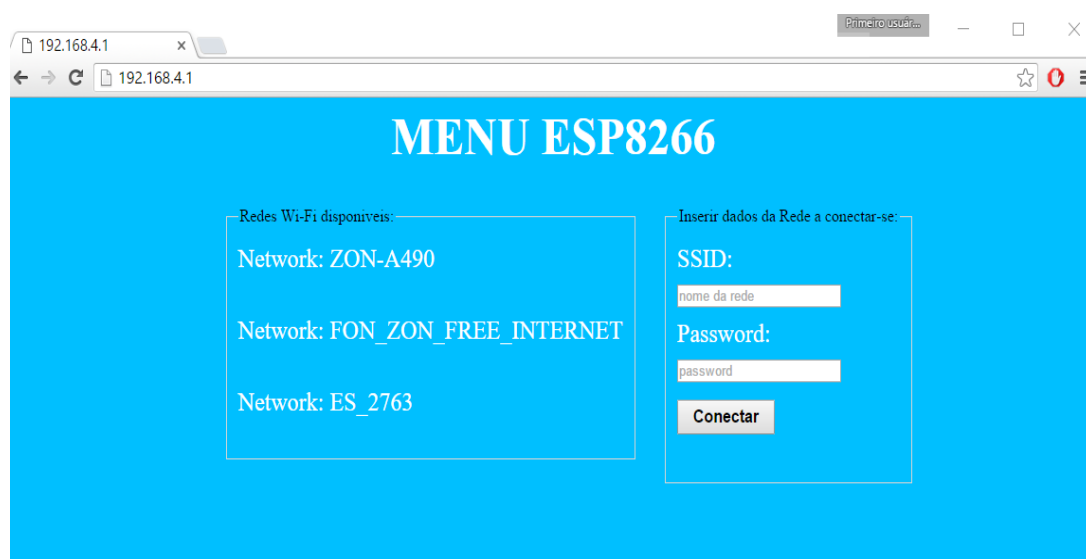


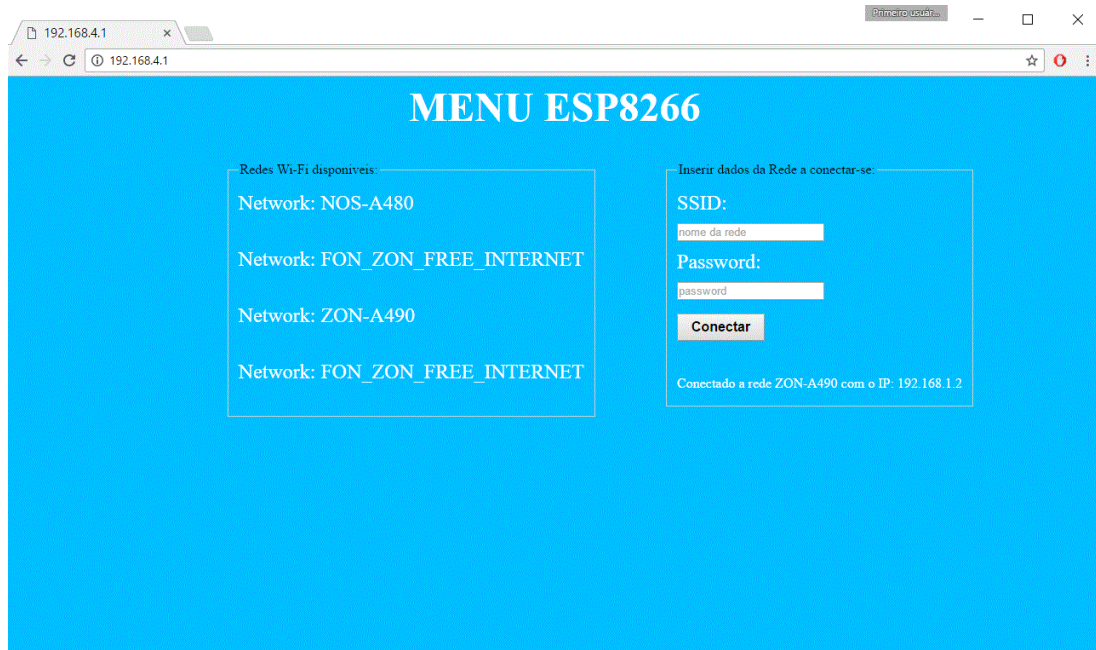
Figura 6.1 – Redes visualizadas pelo PC do Utilizador.

Após realizada a conexão à rede gerada pelo módulo Wi-Fi, o utilizador, para interagir com o módulo Wi-Fi, necessita de iniciar o *browser* e digitar o endereço IP 192.168.4.1, que é o endereço atribuído ao módulo Wi-Fi por omissão. Neste endereço é apresentada uma página Web (Figura 6.2) disponibilizada pelo módulo Wi-Fi. O conteúdo apresentado na página Web permite visualizar as redes nas imediações do módulo Wi-Fi e permite a conexão a uma dessas redes. Para se conectar à rede desejada é necessário inserir o nome da rede (SSID) e a respetiva password da rede a que se pretende conectar o módulo Wi-Fi.



**Figura 6.2 – Menu da página Web do módulo Wi-Fi.**

Quando realizada a conexão com êxito à rede desejada é apresentado o endereço IP atribuído ao módulo Wi-Fi (Figura 6.3), indicando que a conexão foi realizada com sucesso. Estabelecida a conexão à rede por parte do módulo Wi-Fi, este inicia o servidor TCP abrindo a porta 888, permitindo que um utilizador se conecte ao módulo Wi-Fi através do seu endereço IP e da porta atribuída para esse efeito.



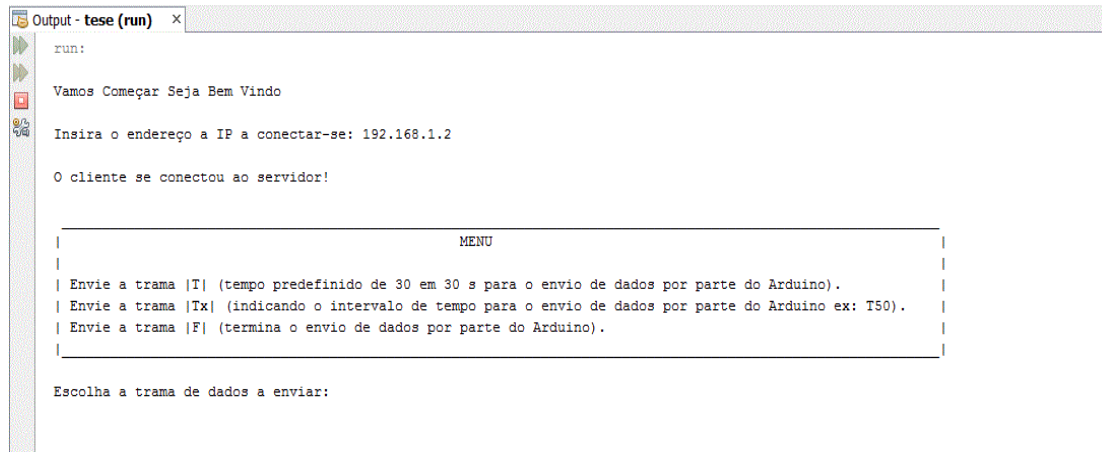
**Figura 6.3 – Conexão com sucesso à rede desejada.**

Com este teste é possível demonstrar o quão fácil é realizar a conexão de um dispositivo que possua a tecnologia Wi-Fi ao módulo ESP8266, e posteriormente a realização da conexão do módulo ESP8266 a qualquer rede que esteja presente nas imediações. Este teste foi realizado com sucesso, permitindo avançar para o segundo teste referente à conexão do Arduino ao módulo Wi-Fi, recorrendo à utilização da interface de comunicação UART para realizar a comunicação entre o Arduino e o módulo Wi-Fi.

## **6.2 Teste à comunicação da plataforma genérica para IoT**

Este segundo teste tem como objetivo demonstrar a comunicação entre todos os componentes utilizados nesta dissertação. Para comprovar o correto funcionamento da arquitetura implementada nesta dissertação, foi utilizada a aplicação para teste descrita na secção 5.1.3. Como descrito anteriormente, esta aplicação tem como principal objetivo permitir realizar conexão ao módulo Wi-Fi, permitindo assim testar o correto funcionamento da rede comunicação desenvolvida. O primeiro passo é conectar-se ao módulo Wi-Fi utilizando o endereço IP que lhe foi atribuído, neste

caso 192.168.1.2, e a porta 888. Na figura 6.4 estão representadas a inserção dos dados e a indicação de que a conexão foi realizada com sucesso entre a aplicação e o módulo Wi-Fi.



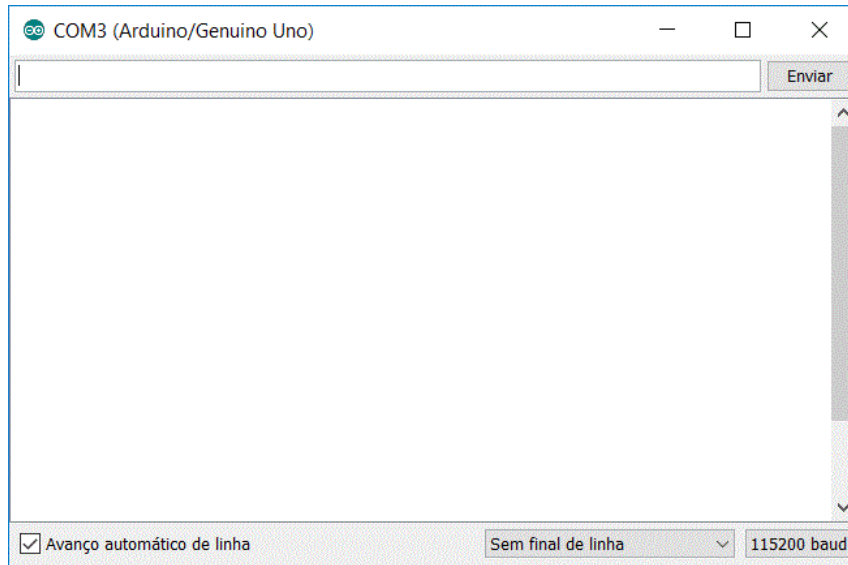
```
run:
Vamos Começar Seja Bem Vindo
Insira o endereço a IP a conectar-se: 192.168.1.2
O cliente se conectou ao servidor!

|-----|
|          MENU          |
|-----|
| Envie a trama |T| (tempo predefinido de 30 em 30 s para o envio de dados por parte do Arduino). |
| Envie a trama |Tx| (indicando o intervalo de tempo para o envio de dados por parte do Arduino ex: T50). |
| Envie a trama |F| (termina o envio de dados por parte do Arduino). |
|-----|

Escolha a trama de dados a enviar:
```

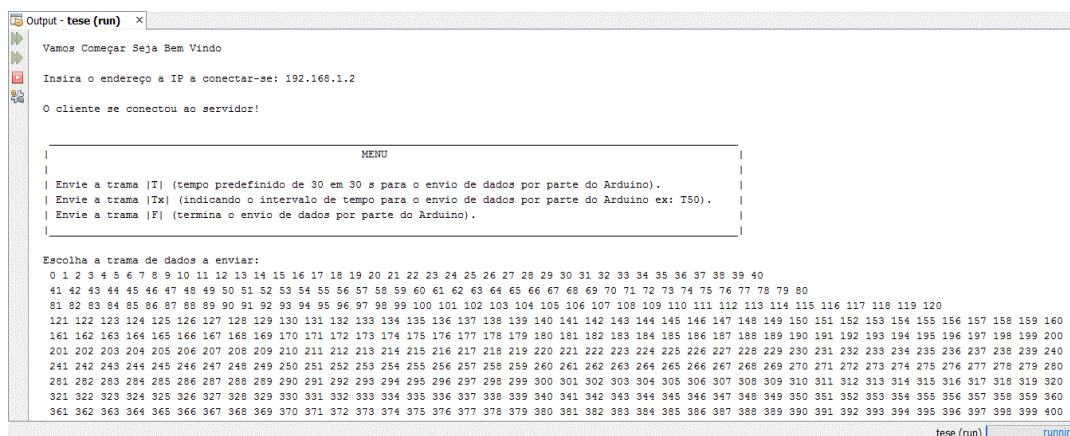
**Figura 6.4 – Aplicação para interação com o módulo Wi-Fi.**

Estabelecida a comunicação com o módulo Wi-Fi é iniciado o teste à comunicação entre os vários blocos que constituem a arquitetura implementada nesta dissertação (figura 4.1). Foi utilizada a interface de comunicação UART para conectar o Arduino ao PC que emula os monitores de QEE, permitindo realizar o teste à comunicação entre os vários componentes da arquitetura. O IDE do Arduino disponibiliza uma ferramenta denominada por Monitor Série (figura 6.5) que permite enviar e receber dados do PC utilizando a interface UART do Arduino.



**Figura 6.5 – Monitor série da IDE do Arduino.**

Com o recurso a esta ferramenta foi possível testar a utilização da interface UART e do protocolo de comunicação desenvolvido para interação com o Arduino. Na figura 6.6 é utilizada a interface de comunicação UART do Arduino, permitindo realizar a comunicação entre o Arduino e o PC monitores de QEE. Para testar o correto funcionamento da plataforma genérica para IOT é utilizado o Monitor Série instalado no PC que substitui os monitores de QEE, permitindo realizar o envio de dados para a aplicação do Utilizador. Na figura 6.6 está representada a receção dos dados por parte da aplicação, demonstrando assim o correto funcionamento entre todos os dispositivos implementados no desenvolvimento desta dissertação.



**Figura 6.6 – Teste à interface de comunicação UART do Arduino.**

O segundo teste foi realizado com sucesso, permitindo demonstrar que o protocolo de comunicação criado para realizar a interação com o Arduino funciona corretamente. Todos os dados enviados através da ferramenta Monitor Série do IDE do Arduino foram recebidos na aplicação do Utilizador desenvolvida para testes, podendo-se avançar para o terceiro teste.

### **6.3 Teste à aplicação que emula o monitor de QEE**

Este terceiro teste tem como principal objetivo testar a aplicação desenvolvida para emulação dos monitores de QEE do GEPE. Para este teste recorreu-se a utilização de dois PCs, o primeiro PC executa a aplicação descrita na secção 5.2.3, sendo que esta aplicação permite realizar o envio de 64 amostras de dados que tem a funcionalidade de serem dados idênticos aos que um monitor QEE do GEPE iria gerar, dados estes fornecidos pelo aluno do MIEEIC. Este PC de emulação (secção 4.2.1) foi ligado ao Arduino utilizando a interface de comunicação UART.

O PC do Utilizador contém a aplicação desenvolvida para o utilizador (secção 5.2.2), que permite realizar a interação com a plataforma genérica desenvolvida nesta dissertação. Para isso é necessário enviar uma trama de controlo contendo uma das várias combinações disponibilizadas na tabela 5.2 para realizar o pedido da respetiva trama de dados ao PC de emulação. Neste terceiro teste utilizou-se a trama do tipo “1”, para receber os valores da tensão e da corrente da fase A. Ativada a interface UART do Arduino, este fica à espera que o PC de emulação inicie o envio dos dados, que são encaminhados do Arduino para o módulo Wi-Fi, que por sua vez envia para aplicação do PC do Utilizador utilizando o Router como intermediário. Isso permite testar a comunicação entre todos os componentes mencionados para o desenvolvimento desta dissertação.

Na tabela 6.1 estão representados os valores numéricos relativos à tensão (VA) e à corrente (IA) da fase A enviados pela aplicação executada no PC, que emula o monitor de QEE do GEPE. De referenciar que na tabela 6.1 só são apresentadas 26



amostras devido ao facto de na figura 6.7 só serem apresentadas 26 amostras das 64 recebidas.

**Tabela 6.1 – Valores recolhidos pelo ADC referentes à Fase A.**

Fase A	
VA	IA
206	809
495	1510
753	2137
1004	2740
1239	3295
1442	3773
1629	4219
1818	4665
2036	5207
2271	5766
2454	6179
2553	6382
2614	6495
2644	6543
2674	6594
2703	6645
2727	6685
2743	6706
2745	6689
2691	6510
2562	6166
2392	5713
2178	5154
1943	4553
1725	4015
1551	809

Na figura 6.7 estão representados os dados recebidos no PC do Utilizador, permitindo visualizar o resultado do teste, onde são apresentados os dados recebidos. Cada linha de dados corresponde ao valor da tensão e da corrente da Fase A, de notar que na primeira linha é apresentado o valor /1/ (tabela 5.2 secção 5) que indica qual a trama de dados que foi enviada do monitor de QEE para o PC do Utilizador. Os valores apresentados já foram convertidos nos valores reais e foram organizados num *array* que permite visualizar os valores da tensão e da corrente para o mesmo instante de tempo.

Para realizar a conversão dos valores enviados pelo PC monitor de QEE são utilizadas as seguintes fórmulas para obter o valor real da tensão:

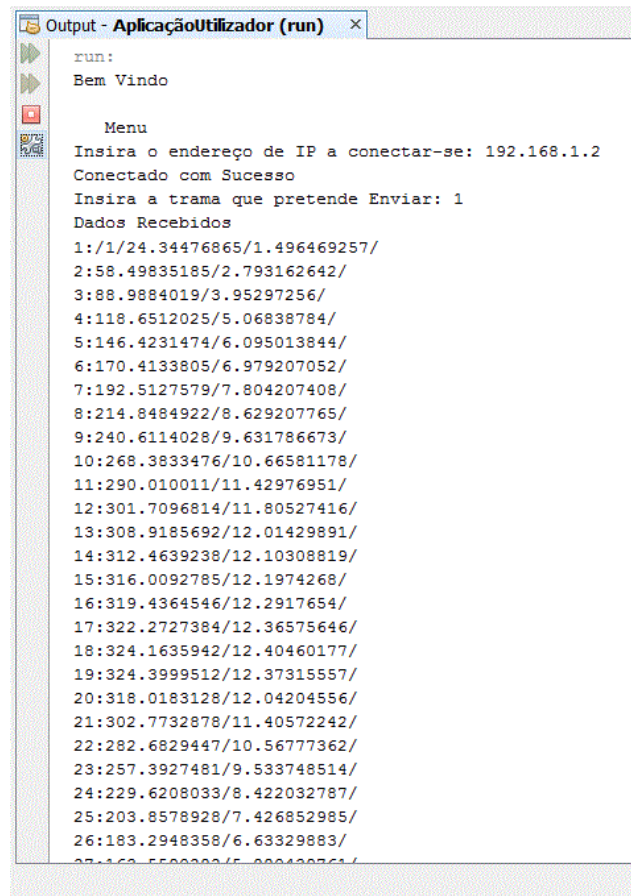
$$\text{Voltage\_factor} = 0,118178489 \quad (1)$$

$$V_{real} = VA * \text{Voltage\_factor} \quad (2)$$

Para obter o valor real da corrente são utilizadas as seguintes fórmulas:

$$\text{Current\_factor} = 0,001849777 \quad (3)$$

$$real = IA * \text{Current\_factor} \quad (4)$$



```
Output - AplicaçãoUtilizador (run) x
run:
Bem Vindo

Menu
Insira o endereço de IP a conectar-se: 192.168.1.2
Conectado com Sucesso
Insira a trama que pretende Enviar: 1
Dados Recebidos
1:/1/24.34476865/1.496469257/
2:58.49835185/2.793162642/
3:88.9884019/3.95297256/
4:118.6512025/5.06838784/
5:146.4231474/6.095013844/
6:170.4133805/6.979207052/
7:192.5127579/7.804207408/
8:214.8484922/8.629207765/
9:240.6114028/9.631786673/
10:268.3833476/10.66581178/
11:290.010011/11.42976951/
12:301.7096814/11.80527416/
13:308.9185692/12.01429891/
14:312.4639238/12.10308819/
15:316.0092785/12.1974268/
16:319.4364546/12.2917654/
17:322.2727384/12.36575646/
18:324.1635942/12.40460177/
19:324.3999512/12.37315557/
20:318.0183128/12.04204556/
21:302.7732878/11.40572242/
22:282.6829447/10.56777362/
23:257.3927481/9.533748514/
24:229.6208033/8.422032787/
25:203.8578928/7.426852985/
26:183.2948358/6.63329883/
27:169.5508002/5.002400761/
```

Figura 6.7 – Valores recebidos do PC de emulação.

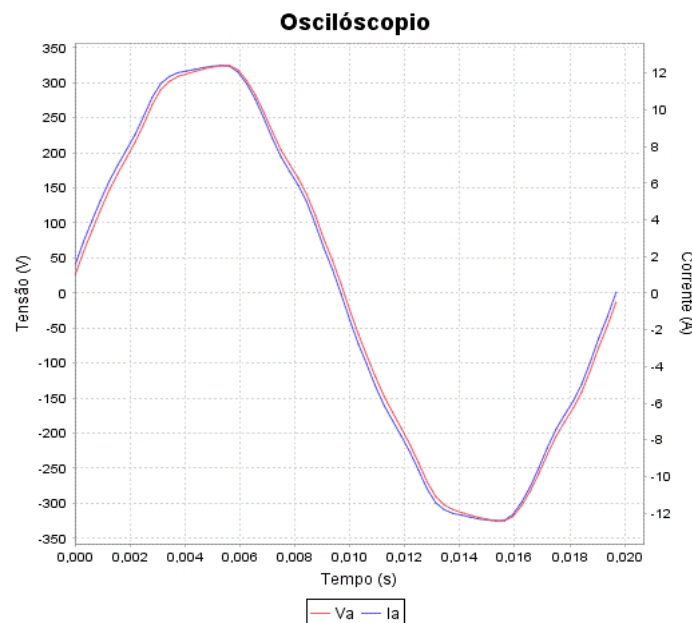
O terceiro teste foi realizado com sucesso, sendo assim possível avançar para o quarto teste, que consiste em testar a aplicação para o GEPE.

## 6.4 Teste à aplicação para o GEPE

Este quarto teste tem como principal objetivo testar a aplicação desenvolvida para PC do Utilizador, descrita na secção 5.2.2. Este teste foi realizado com o recurso a dois PCs, como no terceiro teste, onde o PC de emulação tem como principal objetivo substituir o monitor de QEE do GEPE, de modo a simular dados reais que este pode gerar.

A interface de comunicação usada para realizar a comunicação entre o Arduino e o monitor QEE é a interface de comunicação SPI. No entanto, neste teste a interface de comunicação usada entre o Arduino e o PC de emulação foi a interface UART.

Na figura 6.8 é possível visualizar o resultado para um dos objetivos da aplicação desenvolvida para o GEPE, que consiste no desenvolvimento de um osciloscópio virtual que representa as formas de onda (64 amostras) das diferentes grandezas medidas pelo monitor de QEE. No teste realizado são demonstrados os valores da tensão das três fases em simultâneo.



**Figura 6.8 – Forma de onda da tensão e da corrente da fase A.**

Na figura 6.9 está representado o valor de defasamento calculado entre as duas ondas representadas na figura 6.8 bem como o cálculo da potência reativa, da potência indutiva e da potência total.

O cálculo do defasamento é realizado através da transformada de Fourier, sendo aplicada a transformada sobre os valores recolhidos da tensão e da corrente, permitindo assim calcular o defasamento entre as duas ondas.

A potência aparente é calculada utilizando a seguinte fórmula:

$$P_{ativa} = (\sum_{n=1}^{64} V_n * I_n) / 64 \quad (5)$$

A potência aparente é calculada utilizando as seguintes fórmula:

$$Valor\ eficaz\ (V) = (\frac{1}{64} \sum_{n=1}^{64} V_n^2) \quad (6)$$

$$Valor\ eficaz\ (I) = \left(\frac{1}{64} \sum_{n=1}^{64} I_n^2\right) \quad (7)$$

$$Paparente = Valor\ eficaz(V) * Valor\ eficaz(I) \quad (8)$$

A potência aparente é calculada utilizando as seguintes fórmula:

$$Preativa = \sqrt{Pativa^2 - Paparente^2} \quad (9)$$



**Figura 6.9 – Resultados do desfasamento, da potência, potência reativa e potência aparente.**

Na figura 6.10 está representado o teste realizado ao alarme de sobretensão, permitindo assim visualizar a ocorrência do alarme com a transição do quadrado verde para vermelho, bem como a data e a hora em que o alarme ocorreu.

Na figura 6.11 é possível visualizar o fim do alarme de sobretensão com a representação da transição do quadrado vermelho para verde, bem como a data e a hora em que o alarme desapareceu.

Menu

Conexão Dados

Escolha a opção que deseja pedir:

☐ Tensão e Corrente da Fase A

☐ Tensão e Corrente da Fase B

☐ Tensão e Corrente da Fase C

☐ Tensão da Fase A, B, C

☐ Corrente da Fase A, B, C

Enviar

Alarmes sobretensão	Data	Hora
	15/01/2017	9:50 - PM

Alarme de subtensão	Data	Hora

Alarme de Interrupção	Data	Hora

Figura 6.10 – Representação do alarme de sobretensão ativo.

Menu

Conexão Dados

Escolha a opção que deseja pedir:

☐ Tensão e Corrente da Fase A

☐ Tensão e Corrente da Fase B

☐ Tensão e Corrente da Fase C

☐ Tensão da Fase A, B, C

☐ Corrente da Fase A, B, C

Enviar

Alarmes sobretensão	Data	Hora
	15/01/2017	9:53 - PM

Alarme de subtensão	Data	Hora

Alarme de Interrupção	Data	Hora

Figura 6.11 – Representação do alarme de sobretensão inativo.

## 7. Conclusões

Os sistemas baseados em redes de sensores sem fios têm tido um desenvolvimento promissor devido às necessidades do mercado, nomeadamente na área da saúde, da indústria, das habitações e das cidades. É notório um grande desenvolvimento no que concerne a sistemas que minimizem o consumo energético mantendo a fiabilidade dos dados enviados e recebidos.

Esta dissertação tinha dois objetivos distintos, sendo o principal objetivo o desenvolvimento de uma plataforma genérica para a Internet das Coisas e, numa segunda fase, o desenvolvimento de uma aplicação para o GEPE que permitisse a apresentação gráfica dos dados capturados por monitores de QEE. Inicialmente foi proposto realizar um estudo do estado da arte, com o objetivo de conhecer as várias tecnologias sem fios existentes no mercado, bem como os dispositivos existentes para a formação da Internet das Coisas. Foram estudados em detalhe a plataforma Arduino, bem como o protocolo de comunicação Wi-Fi. Posteriormente foi estabelecida a comunicação entre o Arduino e o módulo ESP8266, e foram efetuados testes de desempenho à comunicação entre o Arduino e módulo Wi-Fi. Foi realizado um teste que tem como principal objetivo comprovar a comunicação entre o Arduino e o módulo Wi-Fi. As principais dificuldades nesta fase do projeto centraram-se no desenvolvimento da programação do módulo Wi-Fi.

Ultrapassada a fase de estabelecimento da comunicação entre o Arduino e o módulo Wi-Fi, procedeu-se à aquisição de dados em tempo real. Para testar toda a plataforma genérica para a Internet das Coisas, recorreu-se a utilização da interface de comunicação UART do Arduino. Foi utilizada esta interface devido ao facto de ser a interface de comunicação que permitia ligar ao PC, permitindo assim testar o envio de dados do PC que emulou o monitor de QEE para o PC do Utilizador.

Para uso da plataforma genérica para IoT foi desenvolvida uma aplicação em ambiente Java que permite ativar/encerrar a receção dos dados recolhidos pelo

sensor que estiver conectado ao Arduino, bem como indicar o intervalo de tempo que o utilizador pretende receber esses mesmos dados. A interface de comunicação a ser usada pelo Arduino depende do meio onde a plataforma será inserida, bem como os sensores que terá acoplados, sendo que a programação do Arduino terá que ser ajustada dependendo da interface de comunicação que se pretenda utilizar.

O segundo objetivo desta dissertação consistiu no desenvolvimento de uma aplicação para realizar a apresentação dos dados recolhidos pelos monitores de QEE. A aplicação consiste na implementação de um osciloscópio no PC com a capacidade de apresentar as formas de onda relativas às fases A, B e C da corrente e da tensão, dependendo dos dados que são pedidos pelo utilizador. Para realização deste processo foi também desenvolvido um protocolo de comunicação que permite realizar a comunicação entre os monitores de QEE e a aplicação desenvolvida para o utilizador, por meio da plataforma genérica para IoT.

Como trabalho futuro propõe-se o desenvolvimento de uma nova aplicação para o utilizador com base na plataforma Android, com o objetivo de receber e enviar dados da plataforma genérica para IoT, permitindo ao utilizador controlar a plataforma em qualquer lugar que esteja utilizando para isso um smartphone.



## Referências

- [1] R. Want, B. N. Schilit, and S. Jenson, “Enabling the Internet of Things,” *Computer (Long. Beach. Calif.)*, no. 1, pp. 28–35, 2015.
- [2] Pedro Castro, João L. Afonso, José A. Afonso, “A Low-Cost ZigBee-based Wireless Industrial Automation System”, 12th Portuguese Conference on Automatic Control (CONTROLO 2016), Guimaraes, Portugal, 14-16 September 2016. pp. 739-749.
- [3] H. F. López, J. A. Afonso, J. H. Correia, R. Simões, “ZigBee-based Remote Patient Monitoring”, Proceedings of 9th International Conference on Wearable Micro and Nano Technologies for Personalized Health (pHealth 2012), Porto, Portugal, June 2012.
- [4] Pedro Macedo, José A. Afonso, Luis A. Rocha, Ricardo Simões, “A telerehabilitation system based on wireless motion capture sensors”, Proceedings of the International Conference on Physiological Computing Systems, PhyCS 2014, Lisbon, Portugal, January 2014, pp. 55-62.
- [5] C. Rowland, E. Goodman, M. Charlier, A. Lui, and A. Light, *Designing Connected Products: UX for the Consumer Internet of Things*. 2014.
- [6] ADEME, “Energy efficiency trends and policies in the EU 27: results of the ODYSSEE-MURE project,” ADEME Editions, 2009.
- [7] J. A. Afonso, F. Rodrigues, P. Pereira, H. Gonçalves, and J. L. Afonso, “Wireless Monitoring and Management of Energy Consumption and Power Quality Events,” Lecture Notes in Engineering and Computer Science: Proceedings of the World Congress on Engineering 2015, London, U.K., 1-3 July, 2015, pp. 338-343.
- [8] E. Systems, “ESP8266EX Datasheet,” pp. 1–31, 2015.

- 
- [9] Pedro Luís Martins da Silva, "Desenvolvimento de um Sistema de Monitorização de Consumos e de Qualidade de Energia Elétrica", Dissertação do Mestrado Integrado em Engenharia Eletrónica e Computadores, Universidade do Minho, 2017., p. 2017, 2017.
- [10] IEEE, *IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, vol. 2012, no. March. 2012.
- [11] P. Baronti, P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta, and Y. F. Hu, "Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards," *Comput. Commun.*, vol. 30, no. 7, pp. 1655–1695, 2007.
- [12] J. S. Lee, Y. W. Su, and C. C. Shen, "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," *IECON Proc. (Industrial Electron. Conf.)*, pp. 46–51, 2007.
- [13] M. Evans, N. Joshua, and J. Hochenbaum, "Arduino em Ação," 2013.
- [14] Atmel, "ATmega48A/PA/88A/PA/168A/PA/328/P," p. 660, 2015.
- [15] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy, "Smart objects as building blocks for the Internet of things," *Internet Comput. IEEE*, vol. 14, no. 1, pp. 44–51, 2010.
- [16] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [17] Belkin, "Belkin - iPhone, iWatch, iPad, Kindle, Samsung & Networking Accessories" (2016). [Online] Página disponível em: <http://www.belkin.com/us/F7C029-Belkin/p/P-F7C029/>.
- [18] Canary, "Canary - A complete security system in a single device." (2016). [Online] Página disponível em: <https://canary.is/canary/>.
- [19] Getpiper, "Piper: All-in-One Wireless Security System" (2016). [Online] Página disponível em: <https://getpiper.com/howitworks/>.

- 
- [20] TCP Connected, "TCP Lighting - Energy Efficient Lighting, like CFL, LED & Fixtures" (2016). [Online] Página disponível em: <http://go.tcpi.com/GetConnected>.
- [21] Neurio, "Neurio - Home Energy Monitor" (2016). [Online] Página disponível em: <http://neur.io/products/>.
- [22] ConnectSense, "Home Automation & Business Monitoring | ConnectSense Wireless Sensors" (2016). [Online] Página disponível em: <https://www.connectsense.com/wireless-sensors>.
- [23] Home Security, "Honeywell: Security for Your Home and Family" (2016). [Online] Página disponível em: [http://homesecurity.honeywell.com/home\\_security.html](http://homesecurity.honeywell.com/home_security.html).
- [24] Iris by lowes, "Iris by Lowe's Simplifies Smart Home Management" (2016). [Online] Página disponível em: <https://www.irisbylowes.com/about/>.
- [25] Securifi, "Securifi" (2016). [Online] Página disponível em: <https://www.securifi.com>.
- [26] HelloIvee, "ivee Voice - Your Voice Assistant for the Home" (2016). [Online] Página disponível em: <http://helloivee.com/>.
- [27] Sense Mother, "Mother Sen.se" (2016). [Online] Página disponível em: <https://sen.se/store/mother/>.
- [28] W. Gateway, "BVW-3653 BVW-3653," pp. 5–6.
- [29] Staples Connect, "Staples Connect" (2016). [Online] Página disponível em: <http://www.staples.com/sbd/cre/marketing/staples-connect/staples-connect.html>.
- [30] SmartThings, "Smart Home. Intelligent Living. | SmartThings" (2016). [Online] Página disponível em: <https://www.smartthings.com/products/hub>.